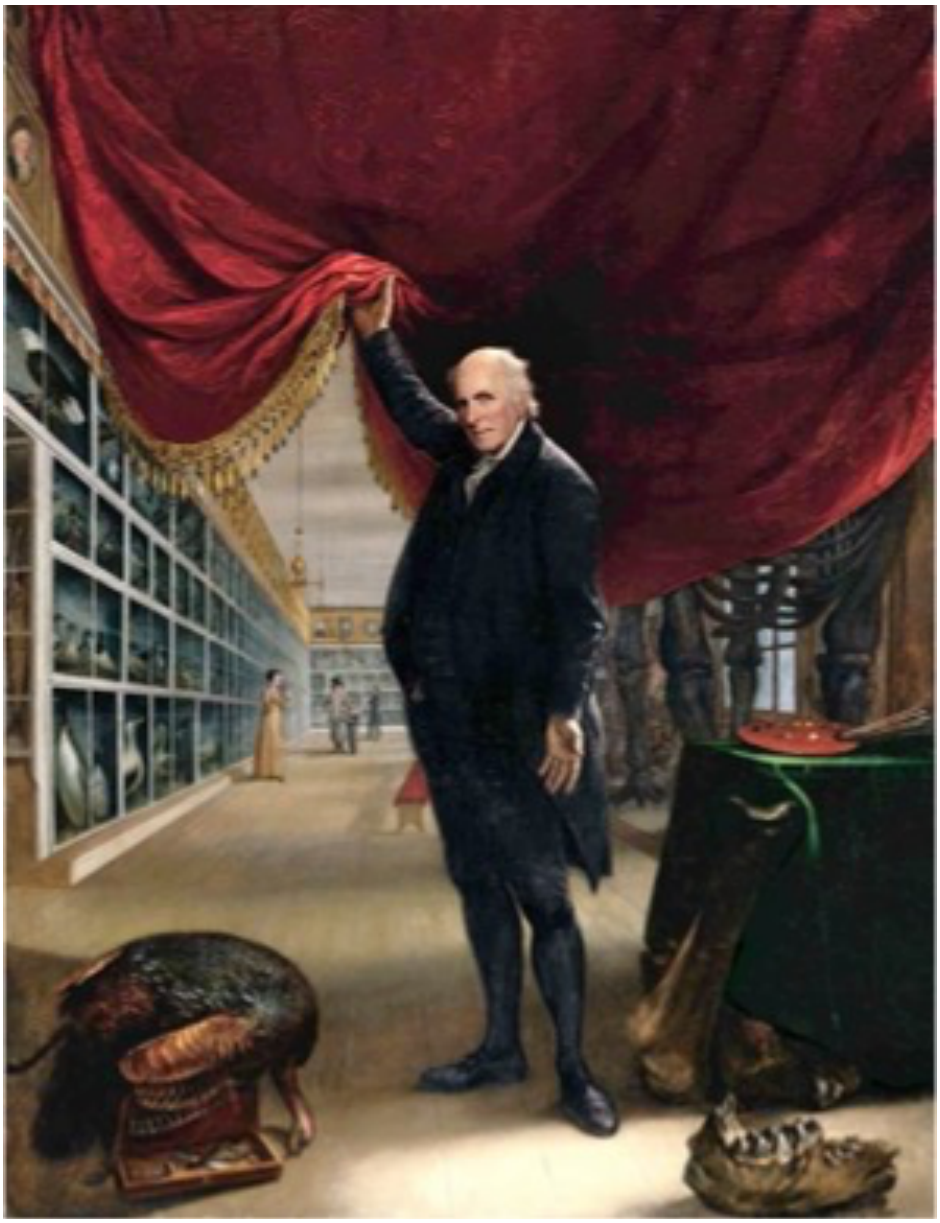
Martin Frické

Artificial Intelligence and Librarianship:
Notes for Teaching

3rd Edition



Martin Frické,
Professor Emeritus
School of Information
The University of Arizona,
Tucson, AZ, USA

ISBN 978-0-473-72294-4



© 2024 SoftOption ® Ltd. (Wanaka, NZ). Email: support@SoftOption.Us



This work is licensed under CC BY 4.0. To view a copy of this license, visit http://creativecommons.org/licenses/by/4.0/

This license requires that reusers give credit to the creator. It allows reusers to distribute, remix, adapt, and build upon the material in any medium or format, even for commercial purposes.

Cover Image: Charles Willson Peale, *The Artist in his Museum* 1822

Author's Note

This intellectual area, and its practical applications, are advancing rapidly. This poses a problem for a book of this kind. Basically, some of it will always be going out of date. Sorry.

7/4/2024

Preface

The third edition has changes and additions. These include:

- a new Chapter 6 on evaluation and the future
- new materials in Chapter 5 on current large language and multimodal models
- scattered revisions, corrections, and updates.

Table of Contents

CHAPTER 1: INTELLECTUAL BACKGROUND 18

1.1 Introduction to Artificial Intelligence 18

1.2 A Genuine Great Leap Forward 24

1.3 Digitization and Transcription 26

1.4 A Paean to Text in Structured Digital Form 29

1.4.1 Text-to-Speech 29
1.4.2 Machine Translation 30
1.4.3 Search and Navigation 32
1.4.4 Preservation and Archiving 33
1.4.5 Free Books! 33
1.4.6 Natural Language Processing 33
1.4.7 Processing by Computer Software 34

1.5 Data and the Need for Good Data 34

1.6 Types of Machine Learning 37

1.6.1 Supervised 37
1.6.2 Unsupervised 39
1.6.3 Semi-Supervised 40
1.6.4 Self-Supervised 41
1.6.5 Reinforcement 43
1.6.6 Reinforcement Learning from Human Feedback (RLHF) 45

1.7 The Concept of Algorithm 46

1.8 Annotated Readings for Chapter 1 48

CHAPTER 2: CHATBOTS 50

2.1 Introduction 50

2.2 Dialog Processing 51

2.3 ELIZA to ALICE 54

2.4 The Turing Test 57

2.5 Machine Learning Chit-Chat Bots 57

2.6 LaMDA 58

2.7 ChatGPT 59

2.8 Task-Oriented 62

2.9 GPTs 65

2.10 Annotated Readings for Chapter 2 68

CHAPTER 3: LANGUAGE MODELS 70

3.1 Introduction 70

3.2 Markov Chains 71

3.3 Hidden Markov Models 75

3.4 Shannon's Guessing Game 77

3.4.1 Introduction 77
3.4.2 Shannon's Approximations as Markov Processes 79
3.4.3 Training a Shannon-Markov Model to Produce 'A Baby GPT' 82

3.5 Taylor's Cloze Procedure 86

3.6 nanoGPT and an Illustration of Training 87

3.7 Embeddings 89

3.8 Word Embeddings and Word2Vec 92

3.9 Adding Knowledge to Language Models 94

3.10 InstructGPT and the Insights it Provides 96

3.11 Annotated Readings for Chapter 3 100

CHAPTER 4: LARGE LANGUAGE MODELS 101

4.1 Introduction 101

4.2 Seq2Seq, Encoder-Decoder Architecture, and Attention 102

4.3 Attention and Transformers 104

4.4 Large Language Models and Foundation Models 105

4.5 Foundation Models 105

4.5.1 BERT 106
4.5.2 GPT-3, GPT-3.5, GPT-4 107

4.6 Bigger is Better and Switch Transformers 109

4.7 Base Models to Assistants to Agents 110

4.8 Concerns and Limitations 117

4.8.1 Hallucinations 117
4.8.2 Fakes and Deepfakes 118
4.8.3 Source Training Data Intellectual Property, Privacy, and Bias 119
4.8.4 Intellectual Property of the Generated Output 121
4.8.5 Cybersecurity 123
4.8.6 Apparent Conflict with Chomsky's Theories 123
4.8.7 Environmental Costs 124
4.8.8 Lack of Transparency 125

4.9 Adding Knowledge and Reasoning to LLMs 126

4.10 Annotated Readings for Chapter 4 127

CHAPTER 5: LARGE MULTIMODAL MODELS 130

5.1 Introduction 130

5.2 Built in Safety Restrictions for GPT-4V 132

5.2.1 'Inherited' Restrictions 132
5.2.2 Privacy 133
5.2.3 Stereotypes and Ungrounded Inferences 133
5.2.4 Be My Eyes— Be My AI 135
5.2.5 An Assessment of the Restrictions 135

5.3 A General Sense of What GPT-4V Can Do 136

5.3.1 Follow Textual Instructions 136
5.3.2 Read Printed or Handwritten Text 137
5.3.3 Read Some Mathematics 143
5.3.4 Read Data and Reason with It 143
5.3.5 Follow Visual Pointing in Images 143
5.3.6 Analyze Images Including Medical Images 145
5.3.7 Use Ordinary Common-Sense Knowledge and Reasoning Across Modes 149
5.3.8 Be an Educational Tutor 150
5.3.9 Use Visual Diagrams When Writing Computer Code 151
5.3.10 Have Temporal and Video Understanding 151
5.3.11 Answer Intelligence Quotient (IQ) Tests 152
5.3.12 Avoid False Presuppositions 153
5.3.13 Navigate Real and Virtual Spaces 153

5.4 Yang et al.'s Conclusion on GPT-4V 154

5.5 GPT-4 Turbo (Early 2024) 155

5.6 GPT-4o (Later 2024) 156

5.7 Google's Gemini 156

5.8 Anthropic's Claude 157

5.9 Meta's LLaMa 158

5.10 Voice 159

5.11 Possible Applications for LMMs 159

5.11.1 Smartphone Uses 159
5.11.2 Spot the Difference 160
5.11.3 Producing Reports from Medical Images 160
5.11.4 Assist with Image Generation 160
5.11.5 Extension with Plugins 161
5.11.6 Retrieval-Augmented Generation (RAG) 161

5.11.7 Label and Categorize Images 162
5.11.8 Identify Objects 162
5.11.9 'Igor', AI Advantage and AI Community 162

5.12 Annotated Readings for Chapter 5 163

CHAPTER 6: EVALUATION AND THE FUTURE 164

6.1 Reliability, Trustworthiness, and Alignment 164

6.2 System 1 and System 2 166

6.3 Benchmarks 167

6.3.1 Introduction 167
6.3.2 Multi-turn dialogs 167
6.3.3 Chatbots 168
6.3.4 Reasoning 168
6.3.4 Common sense reasoning 169
6.3.5 MMLU 170
6.3.6 Coding 171

6.4 Artificial General Intelligence (AGI) 173

6.5 The ARC-AGI Benchmark 175

6.6 Artificial Super Intelligence (ASI) 176

6.7 Annotated Readings for Chapter 6 178

CHAPTER 7: BIAS AND UNFAIRNESS 179

7.1 Algorithmic Pipeline + Data = Machine Learning 179

7.2 Some Clarification of the Terms 'Bias' and 'Unfairness' 181

7.3 Forms of Bias in Wider Machine Learning 186

7.4 Bias in Natural Language Processing 187

7.5 Some Clarification of the Term 'Algorithm' 192

7.6 Computer Program Inadequacy 194

7.7 Bias in the Context of Wider Machine Learning Programs.....197

7.7.1 Fairness ('Distributive Justice') 198
7.7.2 Debiasing Representation 208
7.7.3 Panopticon Bias, the Panopticon Gaze 209
7.7.4 Bias in (Librarianship) Classification 212

7.8 Stochastic Psittacosis: LLMs and Foundation Models 212

7.9 Supplement: The Bias of Programmers 216

7.9.1 The 'Biases' of Professional Programmers 216
7.9.2 The Biases of All of Us as Programmers 218

7.10 Annotated Readings for Chapter 7..... 218

CHAPTER 8: BIAS IN MACHINE LEARNING AND LIBRARIANSHIP...221

8.1 Introduction 221

8.2 Harms of Omission..... 223

8.3 What to Digitize 223

8.4 Search, Primarily Using Search Engines..... 224

8.5 Social Media, Dis-, Mis- and False-Information.....231

8.6 Bias in the Organization of Information 231

8.6.1 Introduction 231
8.6.2 Be Careful, and Sparing, with Emotive Content 233
8.6.3 Warrant and Controlled Vocabularies.....233
8.6.4 The Act of Classification Has Consequences.....239
8.6.5 Taxonomies Have Consequences..... 241
8.6.6 The Current State of Libraries and Their Organizational Systems.....243
8.6.7 Designing Information Taxonomies for Librarianship.....245

8.7 Navigation: Metadata Supported and Otherwise..... 247

8.8 Ethical Arguments to Underpin Assertions of Harms of Bias..... 249

8.9 Annotated Readings for Chapter 8..... 250

CHAPTER 9: WHAT MIGHT NATURAL LANGUAGE PROCESSING (NLP) BRING TO LIBRARIANSHIP? 251

9.1 Introduction 251

9.2 The Pre-Processing Pipeline..... 252

9.3 Text Embeddings and Similarity..... 254

9.3.1 Searching by Meaning (Semantic Search) 256
9.3.2 Research Trails.....257
9.3.3 Classification 258
9.3.4 One Style of Recommendation 258
9.3.5 Plagiarism Detection.....258

9.4 Named Entity Recognition 259

9.5 Topic Modeling 260

9.6 Text Classification Problems 261

9.6.1 Shelving and Subject Classification 262
9.6.2 Sentiment Analysis 262
9.6.3 Author or Genre Recognition.....263

9.7 Controlled Vocabularies, Thesauri, and Ontological Vocabularies . 264

9.8 Indexing and Automatic Indexing..... 265

9.9 Abstracts, Extracts, Key Phrases, Keywords, and Summaries 268

9.10 Text Mining and Question Answering.....271

9.11 Machine Translation.....271

9.12 Evidence 271

9.13 This Is Not Magic 272

9.14 Text Processing and Laws 273

9.15 Annotated Readings for Chapter 9 274

CHAPTER 10: WHAT ARE THE OPPORTUNITIES FOR LIBRARIANS? 275

10.1 Introduction 275

10.2 Librarians as Synergists 279

10.3 Librarians as Sentries 283

10.4 Librarians as Educators 284

10.5 Librarians as Managers 286

10.6 Librarians as Astronauts 287

10.7 Annotated Readings for Chapter 10 288

CHAPTER 11: LIBRARIANS AS SYNERGISTS 290

11.1 Intellectual Freedom 290

11.1.1 Text Recognition 292
11.1.2 Speech to Text 302
11.1.3 Sign Language to Text, and Text to Sign Language 304
11.1.4 Helping Filter and Personalize 305
11.1.5 Scholarly Publishing 306
11.1.6 What Can Be Done With Computer Text 306
11.1.7 ELI5 Translation 306

11.2 Improving the Intermediation Between 'Users' and 'Information Resources'. 307

11.2.1 Some Users Might Not Be Human 307
11.2.2 Some Resources Might Not Be Resources 308
11.2.3 Digital Archiving 308
11.2.4 Enhanced Search Engines 308

11.2.5 Personalization and Recommendation....311
11.2.6 Recommender Systems....312
11.2.7 Understanding What the User is Asking For....315
11.2.8 Text Mining....315
11.2.9 Information Assistants (and 'GPTs')....316

11.3 Improving Traditional Cataloging, Classification, and Retrieval Tools 318

11.3.1 NLP Inspired Improvements....321
11.3.2 Metadata Generation and Automatic Cataloging....322
11.3.3 Some Retrieval Tools....323

11.4 Chatbots....330

11.4.1 Reference Interviews....331
11.4.2 Virtual Services....333
11.4.3 Chatbots as Continuous User Testing of a Library's Public Interface. 334

11.5 Release, Produce, or Curate Training Data....334

11.6 Debunking, Disinformation, Misinformation, and Fakes....336

11.7 Social Epistemology....336

11.8 Robots....339

11.9 Images....341

11.10 Annotated Readings for Chapter 11....342

CHAPTER 12: LIBRARIANS AS SENTRIES....343

12.1 Copyright and Intellectual Property....343

12.2 Intellectual Freedom....343

12.3 Censorship and Algorithmic Curation....344

12.4 Privacy....346

12.5 Bias 347

12.6 Social Epistemology 347

12.6.1 Reliability, Validity, and Over Confidence 347
12.6.2 Confirmation Bias and Poor Reasoning 348
12.6.3 Misinformation 348
12.6.4 Awareness of the Digital Literacy of Patrons 348

12.7 Chatbots 349

12.8 Personalization and Paternalism 350

12.9 Images and Facial Recognition Technology 352

12.10 Losing Jobs 353

12.11 Annotated Readings for Chapter 12 354

CHAPTER 13: LIBRARIANS AS EDUCATORS 355

13.1 Information Literacy (for Consumers of Information) 355

13.2 Artificial Intelligence Literacy 355

13.3 Data Information Literacy (for Producers of Information) 358

13.4 Changes in Learning and Teaching 359

13.5 Scholarly Communication 359

13.6 Academic Libraries Collaborating with other University Units 360

13.7 AI Laboratories in the Library 360

13.8 Automated Decision-Making 361

13.9 Explainable Artificial Intelligence (XAI) 367

13.10 Annotated Readings for Chapter 12 370

CHAPTER 14: LIBRARIANS AS MANAGERS 372

14.1 Coming on Board 372

14.2 Data and Analyses 375

14.3 Evidence-Based Librarianship 376

14.4 Data-Driven Decision Making 377

14.4.1 Collection Building and Management 377
14.4.2 Circulation and User Studies 377
14.4.3 Processing in Libraries 377
14.4.4 Research and Scholarship 378
14.4.5 Service Quality 378

14.5 Acquiring the Appropriate AI Tools 378

14.6 Analysts and Staff 379

14.7 Fear of AI 379

14.8 Annotated Readings for Chapter 14 380

CHAPTER 15: LIBRARIANS AS ASTRONAUTS 381

15.1 Astronaut Training 381

15.2 Why Should You Learn How To Do It? 381

15.3 What are the Real Creative Possibilities 382

15.4 Sitting in Your Tin Can 384

15.5 Exploring World 3 385

15.5.1 Undiscovered Public Knowledge (UPK) 385
15.5.2 Literature-Based Discovery (Text Based Informatics) 388

15.5.3 A Message to Librarian Astronauts 388
15.6 Annotated Readings for Chapter 15 389

APPENDIX A: SOME THEORETICAL BACKGROUND TO LIBRARIANSHIP 390
A.1 Concepts, Classification, Taxonomies, and Items 390
A.2 Controlled Vocabularies, and Thesauri 391
A.3 Ontologies and Ontological Vocabularies 393
A.4 Objective, Intersubjective, and Subjective 395
A.5 Emotive and Descriptive Content 397
A.6 Classification Schemes and the Act of Classification 399
A.7 Annotated Readings for Appendix A 401

APPENDIX B: WORKING WITH LLMS 402
B.1 Introduction 402
B.2 Prompts and Prompt Engineering 403
B.2.1 Basic Examples of Zero-Shot Prompting 405
B.2.2 Examples of Few-Shot Prompting 411
B.2.3 Chain of Thought Prompting 413
B.2.4 Tuning, or Configuring, the Models or Prompts 415
B.3 Choices on Development 418
B.4 Moving Forward With LangChain 421
B.4.0 A Note on the Status of LangChain and Similar as of 11/6/2023 421
B.4.1 What is LangChain? 422
B.4.2 LangChain Experiments Displayed to a Web Page 424
B.4.3 LangChain Using Jupyter 435
B.4.4 Resources for LangChain using Jupyter 438

B.5 Annotated Resources for Appendix B 439

APPENDIX C: TWO IMPORTANT METHODOLOGICAL POINTS.....441

C.1 False Positives and False Negatives.....441

C.2 The Base-Rate Fallacy 443

C.3 Annotated Readings for Appendix C..... 447

APPENDIX D: CAUSAL DIAGRAMS 449

D.1 Causation and Correlation..... 449

D.2 Causal Diagrams 451

D.3 Annotated Readings for Appendix D 467

APPENDIX E: KNOWLEDGE GRAPHS.....468

E.1 Knowledge Graphs 468

E.2 Annotated Readings for Appendix E..... 470

GLOSSARY 471

BIBLIOGRAPHY 507

Chapter 1: Intellectual Background

1.1 Introduction to Artificial Intelligence

Artificial Intelligence (AI) addresses a subset of problems that lend themselves to solution by computers, computer programs or algorithms. Often the AI algorithms are coupled with the use of outside data. The problems in question are ones that an ideally rational and intelligent human being would be able to solve, given the time, resources, and ingenuity. One of the first examples of an AI program within librarianship is that of Optical Character Recognition (OCR). Patrons with poor vision might not be able to read all the books that a library contains. Yet a human library assistant could read books aloud or transcribe them into braille or some other assistive medium. In the mid-1970s, Ray Kurzweil devised and assembled computing machines that did OCR and text-to-speech synthesis. The machines could read books aloud. Kurzweil machines are a massively enabling technology, removing barriers in the access to information. At their core is an AI problem, the problem of recognizing text i.e. classifying letters, and classifying words (where those word tokens might be of different sizes, fonts, colors etc.).

AI in libraries should be distinguished both from automation and from the plain use of computers. Libraries have been taking advantage of automated processes for almost as long as there have been libraries. For example, the Gutenberg printing press, from 1440, is a form of automation; it automates the production of books and documents. Secondly, as society has benefitted

from the introduction of computers, so too have libraries. Computers-in-Libraries is its own wide ranging and active area of study, with conferences and publications. Computers-in-Libraries includes AI uses and more besides. As examples of the more besides, the issuing and tracking of books typically will use computers and databases, as will the financial record systems for the salaries and payment of librarians. AI in libraries is different to automation and plain computing.

Wherein do the differences lie? The use, or simulation, of intelligence, is the main one. What about some further actual and potential examples from librarianship? There is machine translation of texts from one natural language to another. There are 'recommender' systems that can recommend books or resources to patrons. There are, or have been, intelligent assistants in libraries, such as Stella, Beacon, or Bizzy, that can answer reference questions and conduct reference interviews (Sanji, Behzadi, and Gomroki 2022). There are the very familiar search engines and OPACs(Online Public Access Catalogs)/Discovery Systems, most of which have AI components. Final example: AI can add value to the recorded knowledge in libraries by extracting, synthesizing, and summarizing that knowledge. There is an abundance of recorded knowledge in libraries. The sheer volume of the recordings would overwhelm a librarian or team of librarians. Yet, the Lithium Ion 'encyclopedia' was written by a computer program running deep text analysis of holdings within libraries (Writer 2019).

AI has made massive advances in the last 30 years or so, especially in the areas of Machine Learning (ML) and Deep Learning (DL). Both these approaches involve *learning*. Back in 1959, Arthur Samuel wrote:

> … a computer can be programmed so that it will learn to play a better game of checkers than can be played by the person who wrote the program. Furthermore, it can learn to do this… when given only the rules of the game, a sense of direction, and a redundant and incomplete list of parameters which are thought to have something to do with the game, but whose correct signs and relative weights are unknown and unspecified. The principles of Machine Learning verified by these experiments are, of course, applicable to many other situations. (Samuel 1959)

Such programs learn by playing and by associating winning and losing with various strategies, moves, and tactics.

A big advance came about fifty years after this, maybe around 2010, when the amount of data that it was possible to gather, record, and process, expanded massively (amounting to the so-called 'Big Data'). It should be emphasized here that it is not merely the amount of data that is important, but also it is the ability to process that data with a timely throughput (Amodei et al. 2019). Often this kind of learning has similarities to doing statistics with extremely large data sets. ML is distinctly different from other areas of AI, and the difference lies principally with the 'learning'. For example, a while back, computer programs to diagnose a medical condition, say cancer, would mimic expert doctors and consultants, perhaps by attempting to extract rules governing sound or intelligent diagnoses and applying those rules. A programmer, or team of programmers, would write a program to diagnose cancer and the program would be finished and done.

It is true that some tuning might be carried out after the fact, if the program was making some poor or inaccurate diagnoses. Usually, this tuning would consist of the programmers re-writing or changing the program, not of the program modifying itself by learning. So, in general, these programs themselves did not pay a lot of attention to data that arose after they were written. This style of approach is AI by 'expert system'. Nowadays ML would approach diagnoses entirely differently. There would be a 'blank slate' program that could take in vast amounts of data, data about different patients, and their many and varied properties, qualities, and 'features', data about the images, and their properties, and so on. The program would be set up to focus on the 'label' which identified whether a patient had cancer or not. Then there would be a *training set*, which would be data about existing patients, and even members of the public, and known data as to whether they had cancer. The program would learn which features were associated, or correlated, with (the label of) cancer and which not. It would be able to predict which members of the training set had cancer (hopefully with a good degree of accuracy). Then, given success, the program would be released in the world at large and applied to a *test set* or *test cases* i.e. real cases where the diagnoses are not known. (And, possibly, results for real cases will be added through time to the training set to improve performance yet further— although care would be needed were this to be attempted.) You can see how in some ways this differs from diagnoses by medical doctors and consultants or by expert system. The consultants will have had years of training, of course, but they are not in any position to consider anywhere near the number of features that a program can look at. A DL program might consider thousands of features and have been trained on tens of thousands of cases. Also, while some, perhaps most, of the

consultants' knowledge will be explicit, some of it will be tacit (i.e. not articulable). It would amount to perhaps vague unspeakable hunches refined by years of experience. Tacit knowledge is a problem for the prior expert system AI approach: for somehow hunches would have to be converted into rules. But if the consultants themselves cannot put this tacit knowledge into words, how is this to be done? Then, going back to consultants, tacit knowledge itself is hard to produce, instill, and disseminate. If more medical consultants are needed in a health system, the newcomers will have to go through an apprenticeship with an accomplished consultant and years of training. Of course, reproducing a computer program can be done with the click of a mouse. A question is: which system is best at cancer diagnosis? With some forms of cancer, the ML programs provide the best diagnoses (K. Das et al. 2021). In sum, with cancer diagnoses, ML can do better than both prior AI expert systems, and human doctors.

Deep Learning (DL) is a subset of ML inspired by the human brain, in particular by neurons and neural nets. Neurons are nerve cells in the brain that can communicate with other neurons using electrical signals and synapses. Specific neurons are triggered, or become excited, and that activation cascades through other neurons. This provides function to the nervous system— to thinking, reasoning, responding, learning, paying attention, and other cognitive abilities. In a DL implementation, there is a network of software 'neurons' organized in layers. Certain levels of thresholds of input features activate neurons in the first layer; in turn, first layer activated neurons activate some neurons in the second layer, and so on through several layers. At the output level, particular (software) units

indicate, for example, whether the patient being diagnosed likely has cancer. DL is quite involved to set up, and it is demanding on resources. Large amounts of data are needed, then large amounts of training time and computing power to adjust the activation levels and various other 'biases'. Separately, DL often can be opaque as to what is going on. By the time the triggering has gone through several layers, transparency can be lost and that is distinctly a drawback.

Here is an example (Google for Developers 2022). A cancer diagnosis DL program may learn by processing images from different hospitals. But if one of the hospitals is specifically a cancer hospital, then that feature, if used by the DL program, may result in certain types of images being given a higher probability of indicating cancer— those images sourced from the cancer hospital. And the program would be right (those images do have a higher probability of indicating cancer). But, really, this is being right for the wrong reason. You want the DL program to be analyzing the images (supplemented perhaps with facts about the patients and their histories) not reasoning from the originating hospital of the images. To guard against possibilities like these, transparency in the DL software program helps. You need to know what the program is doing, what the program is using in its reasoning. But DL programs can lack transparency.

Two areas of human intelligence had proved special challenges to AI: addressing text (including languages, and translation) and addressing images. Hitherto, computers could not 'understand' or translate natural languages, and they could not recognize and process images, including

videos, as sources of information. Machine learning and DL have changed that: text and images are now fair game for AI.

AI as a scholarly discipline covers many different areas. But for the purposes of the cascade of recent AI developments, and of AI in libraries, we can focus on ML and DL.

1.2 A Genuine Great Leap Forward

A great leap forward came from Transformers, Large Language Models, and Foundation Models. At the end of November 2022, ChatGPT was released to the public. By January 2023, it had 100 million active users. Many more interested observers were aware of its existence— more than 40% of the adults in the United States know about it. It is the fastest growing, and most widely used, software application of all time. There is some history to it. In 2017, Ashish Vaswani and co-authors published the paper *Attention is all you need* (Vaswani et al. 2017) (see also (Huang et al. 2018)). This introduced Transformers. Shortly thereafter there started to emerge Large Language Models and Foundation Models. (What all these are will be explained later in the book.) ChatGPT is a Transformer, and a Large Language Model, and initially it was a fine-tuned version of the Foundation Models GPT-3 and GPT-3.5. (As of 7/1/2024, ChatGPT uses GPT-4o.)

Pretty much any machine learning or deep learning program can be built from a Foundation Model (that is why they are called 'Foundation Models'). Also, the results of systems built using Foundation Models will likely be

superior to any other approach. So, a correct strategy in solving a machine learning problem is to address it using a Foundation Model. But Foundation Models themselves are very expensive, and resource needy, to create. We are talking here of hundreds of millions of dollars, months of computing time, and of using a large portion of the internet as data. Only a few large commercial companies have been able to produce the biggest and best of the Foundation Models. Producing Foundation Models is not the sort of thing that you and I are going to do, nor are most universities, nor even most governments. Some Foundation models have been open-sourced and are freely available to all. This is a mixed blessing. Allowing programmer/users to have the code, lets them see what the code is and, historically, with open-sourced projects like Linux, the programmers can contribute, improve the code, 'catch bugs', etc. But Foundation model ML code is a little different. There are deep security concerns and great potential for unintentional, and even intentional, harm. Trusting a few massive companies like Google, OpenAI, and Microsoft to look after us is not brilliant, but it is probably better than making the code available to all and sundry (including bad actors). That said, the massive company Meta open-sources its code. Also, Hugging Face provides a hub, a library of open-source Foundation Models (Hugging Face 2023). Some commercial Foundation Models have Application Programming Interfaces (APIs) that allow Users to pay a fee and use them (for now at least). For example, May 2023, from (OpenAI 2022c), you can pay $20 a month and have good API access to GPT-4.

From an educational point of view, we can take a machete and cleave out and discard pretty much all of machine learning prior to Foundation Models and start our studies at that point. Andrej Karpathy writes:

> … the whole setting of training a neural network from scratch on some target task (like digit recognition) is quickly becoming outdated due to finetuning, especially with the emergence of foundation models like GPT. These foundation models are trained by only a few institutions with substantial computing resources, and most applications are achieved via lightweight finetuning of part of the network, prompt engineering, or an optional step of data or model distillation into smaller, special-purpose inference networks. I think we should expect this trend to be very much alive, and indeed, intensify. In its most extreme extrapolation, you will not want to train any neural networks at all. (Karpathy 2023b)

1.3 Digitization and Transcription

Digital computers work with electronic digits, surprise. They work with the digits 0s and 1. But, unfortunately, at least some of the information resources that the ML algorithms have the potential to address are not, or were not, in digital form. For example, Shakespeare's only surviving playscript— *The Booke of Sir Thomas Moore*— was not (British Library 2020). So, digitization of those resources not born digital is an important precursor to wide-ranging ML in librarianship.

The 2002 Google Books Project, or Google Project Ocean, was a very early attempt to address digitization. Its approach was to use OCR scanning on the physical resources. It was not prompted by the needs of ML, rather it was aiming for a Universal Library. As James Somers writes:

> You were going to get one-click access to the full text of nearly every book that's ever been published. Books still in print you'd have to pay for, but everything else—a collection slated to grow larger than the holdings at the Library of Congress, Harvard, the University of Michigan, at any of the great national libraries of Europe—would have been available for free at terminals that were going to be placed in every local library that wanted one.
> At the terminal you were going to be able to search tens of millions of books and read every page of any book you found. You'd be able to highlight passages and make annotations and share them; for the first time, you'd be able to pinpoint an idea somewhere inside the vastness of the printed record, and send somebody straight to it with a link. Books would become as instantly available, searchable, copy-pasteable—as alive in the digital world—as web pages. (Somers 2017)

There is another huge potential benefit that arises from the network effects of having many books digitized collectively. The Google Page Rank algorithm, which the core of its very successful search engine for the web, uses links *between* web pages to establish ranking. Something similar, citation indexing, citation counting, and citation ranking, had long been in use in bibliometrics and informetrics working with paper based libraries (Dizikes 2011; Araújo, Castanha, and Hjørland 2021). If books, journals, and other paper-based resources were also to be available digitally, powerful algorithms using linking then could be used in conjunction with massive compute power— search and ranking of actual books in libraries would be able to be improved. So, the promise was of a Universal Library, with improved search and ranking of the sources.

Unfortunately, the Google Book Project got mired in difficulties of one kind or another, mainly legal, and, as of 2023, it seems to have drifted off into

limbo (Somers 2017; Howard 2017). The work does continue, but in a somewhat slower and piecemeal fashion. We cannot pretend that this is a good result, but modern practices have rendered it less of a total disaster than it might have been. Most published books today are written on word processors on computers. Even if they are not produced that way, they are usually in electronic form prior to being printed (if, indeed, they are printed). The authors or publishers may or may not make the electronic form available to the world at large. Authors write books to have them read. Publishers 'print' books to have them read (and sold or licensed). The parties concerned have incentives. So, one way or another, these electronic forms should become available.

Separately, physical documents themselves are perhaps not so important as they once were. The indexed Web is around 60 billion pages (de Kunder 2022), and Google estimated that the number of physical books they needed to scan was around 130 million. The Web, the Internet, is a lot larger than the combined book collections in libraries.

There is a distinction or consideration that should be mentioned here: that between digitization and transcription (or decoding). Were we to use a modern smartphone to take images of Shakespeare's *The Booke of Sir Thomas Moore* we would have the playscript in digital form. But the playscript itself is in letters, words, sentences, and in speeches. It has structure. It is that structure, or some or most of it, that we would like captured digital form. When Google set out on their book project, they did not aim to photograph every page of every book. They aimed to use Optical Character Recognition (OCR) to get to the structured text in digital form as

0s and 1s. They aimed to transcribe to digitized computer text that could be copied, pasted, read aloud, translated to other languages, etc. It is that structured digitization that is the desired goal. ML can usually work with a digital image of a text and do an OCR extraction of structured text. But, nowadays, when authors are writing on word processors, and publishers are using the digitized form throughout the workflow, structured digitization should be available without having use OCR on images.

Many libraries have digital collections, for example the Library of Congress has https://www.loc.gov/collections/ with millions and millions of items. Its collection *Chronicling America: Historic American Newspapers* has over 20 million images (i.e. photographs) of pages of historical newspapers. That is great, but for certain kinds of research those need to be transcribed into the 0s and 1s of computer text. (Very shortly machine learning will be able to do that task in the blink of an eye.)

1.4 A Paean to Text in Structured Digital Form

Having text in digital form is very valuable in terms of what can be done with it. Here are some possibilities. (There may be some overlap between the categories.)

1.4.1 Text-to-Speech

(Computer) text can be spoken aloud. That is one part of what the Kurzweil machines, mentioned above, do. This is not trivial, by any manner of means, but it is also not a challenging AI problem. Obviously, text-to-

speech computer programs need to know the language of the text and how it is to be spoken (its pronunciation).

1.4.2 Machine Translation

Computer text, from a source natural language, can be translated by computer into other languages. This used to be done in an expert systems style of AI. There would be a dictionary between the source and target languages, grammar rules for both languages, and programmers would try to figure out translation rules that algorithms could use to get from one language to the other. The modern approach would be to used ML and DL. This relies on existing (human) translations between texts. These come from a variety of sources; for example, the United Nations provides translations into six languages of all its texts. The Bible has been translated into many languages, although it does have the drawback of having a very specific and distinctive written style. Google Translate uses the DL program 'Google Neural Machine Translation'. At a first cut, this is what it does. There are a number of texts that have been translated, by human translators, from one language to another, say from English to French (and French to English). To translate from English to French the Neural D.L. will find all the existing occurrences of the relevant sentence, phrase, or word, in English and look up how they been translated by the human translators. Then it will favor what seems to be the best translation. Making the latter judgement depends also on what the D.L. has learned, supplementary rules, dictionaries, etc.

The resulting translations are reasonable, but not perfect. (They can be improved by further editing by human readers or translators.) You can see where some of the problems may lie. Context is one. Single words might be homographs (for example, 'bank' (financial institution) and 'bank' (side of a river)) and the translation system might lack the context to disambiguate them. Plain lack of understanding may be another. Human translators presumably will understand their translation, and this would prevent certain kinds of ridiculousness. In a general way, a neural DL will learn what is silly and what is not. But still, in no sense does it understand the text, and this leaves open to making certain kinds of mistakes (i.e. those where the words, grammar, and construction are all fine but the result is nonsense).

Douglas Hofstadter offers interesting examples, including:

> In their house, everything comes in pairs. There's his car and her car, his towels and her towels, and his library and hers. (Hofstadter 2018)

As of 2023, machine translation simply cannot translate this correctly, say from English to French. An anecdote. The author's wife had some dental treatment while living in Japan. The Japanese hygienist had a little device that she spoke into in Japanese, and it spoke back in English. So, this is digitization and transcription of speech, translation of one language to another, then speaking aloud the resulting translation. Here is a real example of what it said to her:

> Pardon me, may I please put red dye on your dirty teeth!

Despite the occasional oddity, machine translation can work 24 hours a day, 7 days a week on a vast number of sources. It is not expensive. In sum, machine translation can produce a bulk of translations of reasonable quality. Progress in this area is rapid for example, Ankur Bapna and co-authors have a paper *Building Machine Translation Systems for the Next Thousand Languages* (Bapna et al. 2022).

1.4.3 Search and Navigation

Once a text is in digital form it can be searched and navigated very efficiently, even if it has no preemptively prepared index. For example, the words 'butterflies' and 'rhopalocera' are synonyms, and 'lepidoptera' is a term for an order of insects that includes rhopalocera (i.e. it is a more general term). Imagine yourself to be a young person with a physical book on the subject of insects. You are interested in the specific topic of butterflies. But perhaps the word 'butterflies' is not in the index of the book, nor does it appear frequently in the text if at all (although 'rhopalocera' appears many times, being the favored word of the synonym pair). Also, on this occasion, you are interested more widely in insects. You would like your search to include butterflies but also, in part, to be a little more general. Basically, you will be disappointed with what you are able to achieve (which will be more-or-less nothing). In contrast, were the book to be digitized and it were searched by computer (either with Large Language Model, or with thesaurus support), the desired tasks and goal would be trivial and satisfiable in milliseconds.

1.4.4 Preservation and Archiving

Loss of the contents of some texts could be catastrophic— for example, contracts, treaties, or the works of Shakespeare. Once digitized, once there is a digital surrogate, there are cryptographic techniques to preserve text surrogates 'forever'. They could be encrypted, hashed, and put into a suitable public blockchain using content-based indexing. Alternatively, one digital copy could be placed in the *Billion Year Archive* through the good graces of the Arch Mission Foundation (Spivak and Slavin 2023).

1.4.5 Free Books!

Once text is in digital form the marginal cost of duplication is near zero. Of course, there are considerations of licensing and intellectual property.

1.4.6 Natural Language Processing

Natural Language Processing (NLP) has always used 0s and 1s and computers. But it has absolutely flourished with ML, and especially with Large Language Models (which we will discuss later). As a selection of possible procedures or techniques, there is: the entire field of digital

humanities, text summarization, text mining, question answering, information extraction, text categorization, sentiment analysis, plagiarism detection, author and genre recognition, word sense disambiguation, and lexical and ontological acquisition, and text analysis for social applications such as blogs and social networks. So, for example, given two physical texts by unknown authors and the question 'are these texts written by the same author?', digitization and NLP can provide the answer.

1.4.7 Processing by Computer Software

Much data either exists or was initially recorded on a physical medium, such as paper or cards. This data really needs to be digitized in order to be processed by statistics or data science. This processing would then amount to data processing, or data mining, or text data mining (TDM). Vast continents of knowledge or information would be opened up.

1.5 Data and the Need for Good Data

ML aims to learn, and what it is going to learn from is data. Primarily this data is the encoding of observational or experiential reports about the world. At one point, historically, such data would have been assumed to be a solid bedrock. Nowadays, most philosophers of science would describe observation reports, or data of this kind, as being 'theory-laden'. Really, they are two aspects to this assertion. Any observation report relies on theories which infuse either the instruments used or the raw observations

(if there is any such thing as truly raw observations) (Duhem 1914; Hanson 1958). Duhem writes:

> Go into this laboratory; draw near this table crowded with so much apparatus: an electric battery, copper wires wrapped in silk, vessels filled with mercury, coils, a small iron bar carrying a mirror. An observer plunges the metallic stem of a rod, mounted with rubber, into small holes; the iron oscillates and, by means of the mirror tied to it, sends a beam of light over to a celluloid ruler, and the observer follows the movement of the light beam on it. There, no doubt, you have an experiment; by means of the vibration of this spot of light, the physicist minutely observes the oscillations of the piece of iron.
> Ask him now what he is doing. Is he going to answer "I am studying the oscillations of the piece of iron carrying this mirror?" No, he will tell you that he is measuring the electrical resistance of a coil. If you are astonished and ask him what meaning these words have, and what relation they have to the phenomena he has perceived and which you have at the same time perceived, he will reply that your question would require some very long explanations, and he will recommend that you take a course in electricity.
> It is indeed the case that the experiment that you have seen done, like any experiment in physics, involves two parts. In the first place it consists in the observation of certain facts… In the second place, it consists in the interpretation of those facts…
> The result of the operations in which an experimental physicist is engaged is by no means the perception of a group of concrete facts; it is the formation of a judgement interrelating certain abstract and symbolic ideas which theories alone correlate with the facts really observed (Duhem 1914).

A consequence of this is that the reports are conjectural, or fallible. Any of them may be mistaken. They are not a bedrock. Second, any collection of data is a selection among what it is possible to observe or measure. The philosopher of science Karl Popper writes:

> ... the belief that we can start with pure observation alone, without anything in the nature of a theory is absurd; as may be illustrated by the story of the man who dedicated his life to natural science, wrote down everything he could observe, and bequeathed his priceless collection of observations to the Royal Society to be used as evidence. This story should show us that though beetles may profitably be collected, observations may not.
>
> Twenty-five years ago I tried to bring home the same point to a group of physics students in Vienna by beginning a lecture with the following instructions: 'Take pencil and paper; carefully observe, and write down what you have observed!' They asked, of course, what I wanted them to observe. Clearly the instruction, 'Observe!' is absurd. (It is not even idiomatic, unless the object of the transitive verb can be taken as understood.) Observation is always selective. It needs a chosen object, a definite task, an interest, a point of view, a problem. And its description presupposes a descriptive language, with property words; it presupposes similarity and classification, which in their turn presuppose interests, points of view, and problems (Popper 1963).

So, the data used in ML is always incomplete.

It is becoming increasingly common in ML circles to use the phrases 'ground truth' or 'ground truths' or 'ground truth data set'. Caution is needed here. No matter how much care goes into gathering and curating the data, it can still be wrong and incomplete.

There are many other pitfalls that can occur sample data or training data even if the data is absolutely correct. The data may be 'unbalanced'. An example that can be used here is that of a test predicting the (unknown) gender of a person from their (known) height. If the training data, the sample, where the heights and genders are known, consists of 95% males

and 5% females, then mathematics will suggest predicting male no matter what the height of the test subject is.

In sum, getting good training data is tricky.

1.6 Types of Machine Learning

1.6.1 Supervised

Supervised learning is learning with a teacher— a teacher who knows the answers. An example will help here: Optical Character Recognition (OCR). OCR is a solution a supervised 'classification problem'. It can look at some text e.g. 'Call me Ishmael' and identify, or recognize, that the first letter of the text is a 'C', the second letter is an 'a', and so on. ML OCR will approach this by being taught how to classify characters. It will be supplied with a *training set*, which will be a reasonable sample of letters and the *correct classifications* of what they are. Training sets are typically large. For example, the well-known and widely used MNIST set, which is a collection of hand-written examples of the digits 0 through 9, with correct identification labels, has around 60,000 entries (LeCun, Cortes, and Burges 1998). The overall OCR technique is an optical one, so it is the features of the sample letters that can be detected optically that will be the input (e.g. size, shape, color, grid arrangement of component dots or pixels, etc.). Then the program will attempt to correlate combinations (i.e. vectors) of these with the correct classification e.g. that a particular sample token character is an 'a'. More than likely, the program will make many mistakes initially.

But either the programmers, or the program itself, will tune various parameters (e.g. weights on the components of the vectors) to improve the classification until it reaches an acceptable level of performance. There is an interesting point to be made here about what are known in statistics as 'omitted variables'. As mentioned, the ML program will start by considering optical input from size, color, pixels etc. But it will then learn which variables of these to include and which to omit. The machine here has an advantage over a human statistician as it has the sheer computing power to run through the alternative possibilities in a reasonable time.

The training set needs to be adequate for the task. For example, if the letter ‘j’ does not appear in the training set, it is unreasonable to expect the ML program to classify js correctly. Even if js appear, there needs to be enough of them in the various fonts and scripts (cursive or not, monospaced or proportional, etc.) for the program to be able to learn what is correct and what is not. OCR, i.e. recognizing the actual individual characters, would not usually be an end in itself. Rather, the interest would be in the words that those characters form, or, more generally, the text.

If the OCR, or Text Recognition, application has access to a wider context of text, that can improve its performance. For example, if the ML is recognizing entire words from their component characters, and separators, then the first letter of the word ‘On’ is going to be the letter upper case ‘O’ and not the numeric letter zero ‘0’— the number zero makes no sense in that context.

Supervised learning needs labeled data for its training data. Getting such data with quality and in quantity is not easy. Often experts will be required to do the labeling e.g. of medical images, or of the correct cataloging numbers for books in libraries. But many experts would struggle if confronted with 100,000 items to label. Sometimes the process can be eased, at a financial cost, by crowdsourcing (Wikipedia 2023c). For example, the LaMDA models of dialog agents uses crowdsourcing, where the members of the crowd are supplied with calculators and access to a web browser and a search engine so they can check how reliable the output information is that the dialog agents supply (Thoppilan et al. 2022).

1.6.2 Unsupervised

The word 'supervised' comes in to qualify learning because it is known, for most of the cases, what the correct answers are. A Machine Learner could, alternatively, be challenged to classify marks (characters) on paper into groups of marks (characters) similar to other characters i.e. clustering (forming clusters of similar characters). That would be *unsupervised* learning, where there are no known right or wrong answers (and no teacher).

A first issue here is that there is no ML constraint as to how many clusters there might be, or should be, in any given problem. For example, the music streamer Spotify plays songs that the listener wishes to listen to; Spotify can cluster these songs into similar songs and perhaps mark those as being 'playlists' or 'radio stations'; then the User can listen to a radio station while

cooking, or while going for a walk etc. But the clustering algorithm needs to know 'how many radio stations should there be: 2 stations, 7 stations, 300…?' There is no answer to that question from within the ML system. Spotify itself, or the User, or some outside party, has to decide on the number of stations. Going back to OCR and attempting to do it in an unsupervised fashion. We could help it by saying we would like 36 clusters (we have in mind here one for each letter of the alphabet, and for each digit). Clustering might then lead to mistakes like characters that we consider to be upper case 'I's being put together with lower case 'l's or zeros '0' with upper case 'O's. There is no training set. There are no right or wrong answers available to assess the program. Unsupervised classification might be perfect in certain areas, for example, clustering songs into playlists, but it is not really suitable for OCR.

1.6.3 Semi-Supervised

Supervised and unsupervised classification approaches can be combined. This might be useful when there is a huge amount of data, of which only a small proportion is 'labeled' (i.e. it is known what those items are), and the process of labeling is expensive or time consuming or hard to do. For example:- imagine some historical biodiversity researchers who collected specimen samples in the style of Darwin; they also did as much labeling as they could manage, including labeling at least one example of what they thought was every species they came across; then the initially unlabeled specimens were later to be donated to many museums, who, of course, wanted them labeled; this problem might be approached by clustering,

supplemented with a back-and-forth with supervised learning; then the final labeling of the museum samples could be done by machine.

1.6.4 Self-Supervised

The learning techniques mentioned to date have problems and issues. Supervised learning required large amounts of labeled data which is often difficult, expensive, or even near impossible, to obtain. The need for the quality labeling is the cause of the problem. Unsupervised learning simply might not give you what you want.

Self-supervised Learning (SSL) is an ingenious idea which will often be far superior to its alternatives. Basically, it uses unsupervised learning, and the data itself, to label the data, then it uses supervised learning on the now labeled data. To do this the data has to have suitable structure or patterns in it. This gives a context, or contexts to items of data, and the general problem being addressed needs to be tightly specified or understood. SSL really found its place in Natural Language Processing (NLP), and some examples from NLP may make the technique clearer. The foundation models BERT and GPT-3 both have the pre-training task of predicting the next word, or previous word, from a sequence of words in a passage of English (or other natural languages) (Devlin et al. 2019; Brown et al. 2020). The way they do it is to scan vast amounts of English text e.g. trillions of word tokens such as the entire of the Internet (including Wikipedia, Reddit linked sources, all freely available digitized novels, etc.). Apparently GPT-3

was pre-trained on 45 terabytes of data. This is about the same size as one quarter of the holdings of the Library of Congress. This pre-training provides the various transition probabilities from prefix, or suffix, words or sequences, to the current target word, and, essentially, the solution. Now, the data itself, the English text, is not labeled, so this is not supervised learning. But the scanning of the text can produce a pseudo-label for the missing 'gap word'. For example, the label for 'The cat sat on the <?label?> ' can be produced merely by looking through a vast amount of real-life text. Of course, there does not have to be a single one-word answer to this. 'mat', 'table', 'floor', etc. might all be possible answers. But then there will be probabilities associated with the possible answers, and the wider context will provide guidance.

Self-supervised learning has an obvious home with natural language. But it also can be used with images. There is a context to which patches of colors or pixels are close to other patches. Further, foundation models like GPT-3, GPT-4, etc., are becoming multi-modal. ‘Multi-modal’ means that they can work with different ‘modes’ such as with text, images, audio, video, etc.. The earlier technique for this was to use text as a stepping-stone. The model would be pre-trained, using SSL on text, then fine-tuned, perhaps with some prompts or labeling, to work on images. Nowadays, many Foundation Models can work with different modes natively, without using text as an intermediary step. SSL offers freedom. Getting good, labeled data at scale is difficult, if not near impossible. It is a barrier or bottleneck. But with SSL, it is not needed.

1.6.5 Reinforcement

Reinforcement learning is familiar to us in daily life. It involves exploration of an environment by trial-and-error, and, as part of this, having what are called 'delayed rewards' (Sutton and Barto 2018). The rewards provide feedback as to how well the trial-and-error is working. Imagine a student backpacker having temporary employment picking apples in an orchard. She gets paid for each apple she picks (but for each apple she picks there will be one less apple to pick on the tree that she picked it from). Also, she gets a bonus for each basket of apples that she picks, especially if she fills the basket faster than other pickers. Bigger apples will fill a basket quicker, but there will be fewer of them in the basket. We will assume here that she is trying to maximize her pay, i.e. her rewards. Quite what her best picking strategy might be is a bit of a question. She presumably will have to change trees from time to time, but changing trees is not actually picking. It amounts to dead time invested in the hope of higher rewards later. She will also have to pay attention to other pickers, and to the simple nature of the individual trees. If she works in the orchard for several days, she should be able to learn of a reasonable strategy by trial-and-error i.e. by trying a few approaches and favoring those yielding higher total rewards. The whole process amounts to reinforcement learning.

Reinforcement learning can be used when the ML system has to make a sequence of tasks, or moves, or steps towards some desired goal and there are rewards or penalties for successes or failures. While learning, the system will typically be allowed to try the overall task of reaching the goal

many times. Games are a good setting for this e.g. Tic-Tac-Toe or Checkers or Chess. There will be an environment, and possibly permitted moves governed by rules. The system will be rewarded or penalized according to whether a particular move is judged to be good and also, perhaps, whether the system wins or loses the game overall. The system is not programmed with any strategy or tactics, rather it explores the game by trial and error. Another example of reinforcement learning would be a robot exploring an environment e.g. finding a route through a maze.

Reinforcement learning is not supervised learning— there is no labeled data. It also is not unsupervised learning or self-supervised learning. It is a *sui generis* ML technique— one that needs to have some reward structure. It certainly can be used in language settings. For example, an ML program might be configured to produce five different translations of the same text; then if these could be ranked, perhaps by a human judge; then the rankings could be used as a reward structure, and a reinforcement learning system introduced improve the system at translating. Sometimes, in this context, the reward structure is called the 'reward model'. Typically, reinforcement learning is very compute intensive— e.g. for chess, the ML system may need to play hundreds of thousands of games. There are many algorithms to produce reinforcement learning, but few, if any, are efficient in really large settings. Supposedly, one of the technologies that enabled some of the uses of Foundation Models, such as ChatGPT, was the invention of Proximal Policy Optimization (OpenAI 2017). Proximal Policy Optimization is a reinforcement learning algorithm.

1.6.6 Reinforcement Learning from Human Feedback (RLHF)

Modern Foundation Models or Large Language Models often use Reinforcement Learning in a very specific way. The training of these takes place in two stages: the initial training to produce a plain vanilla base model, then *Reinforcement Learning from Human Feedback (RLHF)* is used to yield the desired product. A textual base model might be produced by self-supervised training on most of the text on the Internet. This might take months to do, and the result might be able to write presentable English. But the model might at that point lack some desirable qualities (such as answering questions given by prompts) and might possess some undesirable properties (such as lying, giving poor medical advice, revealing private information). The model will then be tuned using RLHF. A jury of perhaps 40 people will be assembled and given maybe 10,000 samples of pairs of answers from the model. Each member of the jury will evaluate the answers, saying which of a pair is better and which worse. Then, those evaluations will be fed back to the model as additional training. The model will adjust itself accordingly.

One point of importance about RLHFs is that as the models get really big, as they are going to, RLHF becomes impossible as a practicality. If a model were to produce a billion responses, no human jury would be able to evaluate them. In 2024, RLHF has been one difference between success and failure with Foundation Models. But it is not going to work in its present form going forward.

[There are plenty of companies advertising on the Internet for (paid) RLHF jury participants. Try ‘advertisements for RLHF participants’ in a Google or Perplexity search.]

1.7 The Concept of Algorithm

Programmers, mathematicians, and computer scientists regard algorithms as finite sequences of steps or instructions to perform a computation. The instructions themselves are considered to be atomic, or not in need of further breakdown or explanation. The earliest algorithms are from about three and a half thousand years ago. Here is one from that period to do division of integers by repeated subtraction. It is assumed that we can determine whether one number is smaller than another, that we can subtract one number from another, and that we can count the number of subtractions that we have done. Then the divisor is repeatedly subtracted until the remainder is less than the divisor. Finally, the quotient (the 'result') is the number of subtractions that have been made. For example, to determine the quotient when 22 is divided by 4 (which we all know to be 5):

22-4 = 18
22-4-4 = 14
22-4-4-4 = 10
22-4-4-4-4=6
22-4-4-4-4-4 =2 /* and 2 is less than 4*/
5 subtractions have been made so the quotient when 22 is divided by 4 is 5.

[This algorithm is taught in 3rd grade school mathematics.] The act of programming is using a computer programming language to assemble instructions into algorithms. Usually, this process will be modularized. Small algorithms will be created, as components or modules. Then these algorithmic modules will be combined to produce a complex artifact. The end result might be described as being an 'algorithm', but, given sufficient complexity, the plain moniker 'computer program' might be more suitable. Google's Internet Services consist of 2 billion lines of code; Microsoft's Office Suite for the Mac consists of 30 million lines of code (Desjardins 2017). It would be very strange for programmers or computer scientists to describe either of these as being 'algorithms'. They are computer programs or collections of computer programs.

The word 'algorithm' has become transmogrified in recent times. The vernacular has taken a different turn. Where there is no apparent human agency in a computer program, or computer supported service, that affects humans, folk can and do ascribe the agency to an 'algorithm' or 'algorithms'. For example, nowadays a mortgage company will typically use data, computers, and algorithms to determine whether we might qualify for a loan. If we are turned down, we might well say that the mortgage company's 'algorithm', *the* algorithm *itself*, declared us ineligible for one of their loans. This attribution is not quite right, we will go into that further in Section 7.5.

There is an important detail in connection with algorithms in the context of ML. It is that algorithms can change or self-adjust or self-adapt when faced with (training) data. The ML programs learn by changing their algorithms.

The same beginning, template, or seed algorithm when confronted with two different sets of training data can produce two different trained outcome algorithms. Perhaps that is not a surprise. One would expect different learning from different data to learn from. This adaptation is a small change from how historical algorithms typically behaved, and it can mean, in particular cases, that it is unclear or unknown as to which algorithm is actually running after learning. In turn, this can shroud some or many of the processes in a fog (i.e. in black boxes). This is exactly the opposite of what desirable. Transparency is valuable both for ethical reasons (for example, when making decisions involving humans) and for technical reasons (for example, when trying to assess the correctness of the programs). Part of the lack of transparency comes from the plain complexity of the systems. Some of the ML systems might have a billion parameters or more— then it becomes hard to know what the values for the parameters are and how they interact. There is another word or label that gets used in this context and that is 'model'. When an ML system is trained and its initial algorithm adapts and changes into another algorithm, known or unknown, transparent or hidden, the end result is often called the 'model'.

1.8 Annotated Readings for Chapter 1

Coding Vidya. “Home | Coding Vidya - Best Computer Science Portal -,” 2023. https://codingvidya.com/. (Coding Vidya 2023) Worth a scan. It does, for example, have lists of the Best Online Courses in Machine Learning.

gwern. “Douglas Hofstadter Changes His Mind on Deep Learning & AI Risk,” 2023. https://www.lesswrong.com/posts/kAmgdEjq2eYQkB5PP/douglas-hofstadter-changes-his-mind-on-deep-learning-and-ai. (gwern 2023) Hofstadter is trusted,

knowledgeable, and intelligent. He was an influential critic of AI and ML. No more.

Sanderson, Grant *But what is a neural network? | Chapter 1, Deep learning* (Sanderson and 3Blue1Brown 2017a) and the references he provides. Neural networks have been mentioned in this Chapter, but they have not been described or explained. They are the core of modern machine learning. However, we are constrained by space in this text. Also, explaining neural networks using just written text and diagrams is not the easiest. The medium is not good for capturing the dynamics of it. Far better is an animation or a video. There certainly are some excellent ones available on the web.

Wikipedia. "Algorithm." In *Wikipedia*, 2022. https://en.wikipedia.org/w/index.php?title=Algorithm. (Wikipedia 2022a)This provides useful background on the concept of algorithm. We will have more to say on algorithms in Section 7.5.

Chapter 2: Chatbots

2.1 Introduction

The word 'chatbots' is commonly used in AI as a label for Dialog Agents. In turn, dialog agents might be chit-chat agents or task-oriented agents (Daniel Jurafsky and Martin 2021). Chit-chat agents are understood to be software systems that can 'chat' with you. They chat by interchanging text messages or by using sound and speaking and listening. There is no special purpose to these chats, apart from entertainment, engagement, and perhaps companionship. Task-oriented dialog agents also have interactive sequential conversations. But they can do more (and do less). For example, in a general setting, they can book an airline ticket, order a taxi or a meal, set an alarm or timer, produce a recipe for a birthday cake, retrieve information resources, offer instruction etc. Most everybody is familiar with Apple's Siri, or Window's Cortana, or Amazon's Alexa, or OK Google, which are such agents. These are used on many devices including, most obviously, smartphones.

The task-oriented agents can do less in the following sense. They are directed to one task, or sometimes more, and this really limits the possibilities they have to face. For example, an airline ticket agent needs to be able to 'chat' about departures, destinations, and flight times, but it does not need to be able to chit-chat about the novels of Toni Morrison, last night's baseball game, and how to cure a shank at golf.

There is a suggestion that chatbots may partially replace browsing of certain kinds of web pages. For example, a library home page may have a lot of information on it about the various services or resources that the library offers, navigation through this may be eased by assistance from a chatbot.

2.2 Dialog Processing

> Among the hallmarks of natural dialogs are unexpected and seemingly unpredictable sequences of events. (Bobrow et al. 1977)

It is valuable to have some appreciation of what a dialog is, and how difficult it is model it using software and Natural Language Processing (NLP). We will make simplifications here.

A dialog is a communication that takes place between two or more participants. For us, this can be two participants: one of which is a human, and the other a computer (or software system). At the beginning of the dialog, one of the participants has the *initiative* or *control*. This means that they set the initial agenda, assumptions, and common understandings that are required for the dialog to get underway. There are different conversational settings, as examples, talking about Xmas gifts for children has different assumptions to that of discussing the best recipe for a bundt cake, or to that of booking an airline ticket. Of course, a single conversation can range over many settings. As a dialog progresses, it may be that the agent with the initiative retains it throughout. An example here would that

of a commercial telephone answering and routing system. The caller can make choices among the options on offer, but the caller cannot ask questions or offer choices to the system. More usual are dialogs of *mixed initiative,* where control is sometimes with one agent and sometimes with the other. This usually would be the case with airline ticket reservation systems. The participants can take turns at having the initiative, but also one can retain the initiative through several interactions (for example, with a sequence of yes/no questions). How the initiative might be transferred from one to the other is a complex array of possibilities. There are a number of explicit conversational practices, but also there are implicit conventions (e.g. pauses). Also, an important consideration is whether there are visual or non-verbal cues (as there might be with videoconferencing). There are indexicals, such as 'I', 'now', 'yesterday', etc. Just what these might refer on a particular occasion to is a bit of a question. The computer saying 'I' will have a different reference to the human saying 'I'. But also the indexicals might change their reference as the conversation progresses and the initiative changes. 'Three days earlier' might refer to different times at different points in the conversation. Sometimes the notion of anaphora is introduced here. Anaphora is where the interpretation or reference of one expression depends on another; for example, in 'John Smith is conscientious. He often works evenings.' the word 'he' is anaphoric for 'John Smith'. The computer can avoid anaphora (perhaps at the cost of having a somewhat stilted and unusual conversational style) by always making the references explicit (e.g. saying 'John Smith' instead of 'he'). However, there is no controlling what the human might do.

Further, there is conversational implicature (studied extensively by the philosopher H.P. Grice) (Davis 2019). This is where speaker meaning is different to sentence meaning (and similar related phenomena). Consider an example from (Davis 2019):

> Alan: Are you going to Paul's party?
> Barb: I have to work.

The meaning of the sentence that Barb says is that she has to work. The sentence *itself* does not have any kind of meaning like 'not going to a party'. However, her use of this sentence on this occasion means she is not going to Paul's party. There is a conversational implicature to that effect. So, for example, if a chatbot is collecting RSVPs to Paul's party, it needs to understand 'I have to work' as meaning 'I am not coming'.

There are sentence fragments. For example, answers to questions might be single words or phrases, and grammatically they might be almost anything (nouns, adjectives, verbs, noun phrases etc.) From a logical point of view, we would expect a linguistic component to be either a reference to an item or be a true or false statement about items referred to. But extracting what is what from a fragment might require looking back through to dialog to earlier questions or context. (Bobrow et al. 1977) have the example:

> Computer: In what name should I make the reservation, please?
> Client: Dan Bobrow

Now the 'Dan Bobrow' is a reference to a person, and not a sentence or statement, but what is happening here is that the assertion 'You should

make the reservation in the name of Dan Bobrow' is being made, but to determine that the question and earlier context are needed.

In sum, dialogs are not easy to process.

2.3 ELIZA to ALICE

ELIZA was introduced by Joseph Weizenbaum in the paper *ELIZA—a computer program for the study of natural language communication between man and machine* (Weizenbaum 1966). ELIZA is the first chatbot. It is an early, indeed very early, AI program. However, it is not a ML program. As we will see shortly, it works by rules and pattern substitution. We have an historical interest in ELIZA. Also, its way of manipulating strings, playing with form not meaning, is something of style common in modern machine learned natural language processing. Here is a transcript of a typical conversation with ELIZA:

> Men are all alike.
> IN WHAT WAY
> They're always bugging us about something or other.
> CAN YOU THINK OF A SPECIFIC EXAMPLE
> Well, my boyfriend made me come here.
> YOUR BOYFRIEND MADE YOU COME HERE
> He says I'm depressed much of the time.
> I AM SORRY TO HEAR YOU ARE DEPRESSED
> It's true I am unhappy.
> DO YOU THINK COMING HERE WILL HELP YOU NOT TO BE UNHAPPY
>
> <and so on. The human correspondent's utterances are shown in lower-case, ELIZA's replies in upper-case. This extract is from (Weizenbaum 1966). >

As Weizenbaum tells us, the conversation style here intended to be between a patient and a Rogerian psychotherapist. There is an important sleight of hand in this. A Rogerian psychotherapist feeds, or replies, back some variation of what the patient has said. The psychotherapist does not say anything that requires knowledge of the world. This matters because ELIZA does not then need any knowledge. So, this might be a conversational fragment:

> I went for a long boat ride
> TELL ME ABOUT BOATS

But a request for knowledge would meet with avoidance, perhaps:

> What is a boat?
> HAVE YOU ALWAYS BEEN INTERESTED IN BOATS?

The basic computational procedure for ELIZA is simple. The input text is inspected for one or more matches to patterns. A pattern here usually amounts to a string, where some parts are literals and other parts are variables. So, for example, if we use (?X), (?Y), etc. as variables, then

> I feel (?X)

is a pattern (the 'I feel ' part is a literal, and the '(?X)' is a variable). The matching process takes some input text and a pattern, or patterns, and sees whether they can be matched. For example:

> 'I feel (?X)' can be matched with 'I feel sad' (or, indeed, 'I feel sick', 'I feel happy', 'I feel cold' etc.) but not matched with 'I read novels'

Then there are rules which dictate how the matched results are to be transformed. For example:

> Rule #3: 'I feel (?X)' is to be transformed to 'DO YOU OFTEN FEEL (?X)' where, of course, the value for (?X) is substituted back in.

Usually, several different patterns will match the input text. Then each of those patterns might have several different transformation rules that apply. What happens here is that there is a randomizer which makes a choice of what to use. The reason for this randomization is that it allows ELIZA to produce different responses to the same input text. So, each conversation, or session, will usually be slightly different to earlier ones. At this point, ELIZA can be improved by adding some memory of the conversation or correspondent, being more sophisticated with the responses, and so forth. There are ELIZA like chatbots alive and well today— most notably ALICE (Artificial Linguistic Internet Computer Entity) (Wikipedia 2022c).

There are two important points to note about ELIZA systems. What they do is trivial substitution of one piece of text for another. Merely by looking at the source code and a transcript for a session you can see which patterns and rules were used, step by step. The inner workings are crystal clear. There is no mystery here, nor any intelligence. Nevertheless, a good proportion of folk think that ELIZA systems are intelligent, or even that, when talking with ELIZA, that they are conversing with a human.

Weizenbaum noted this in his original paper, and in 1972 the ELIZA system PARRY passed the Turing Test (i.e. fooled some people into thinking it was human) ((Weizenbaum 1966; Colby et al. 1972)

2.4 The Turing Test

Are you wondering what the Turing Test is? Here is ChatGPT's explanation of it (provided on 12/6/2022, while ChatGPT was available as a research preview):

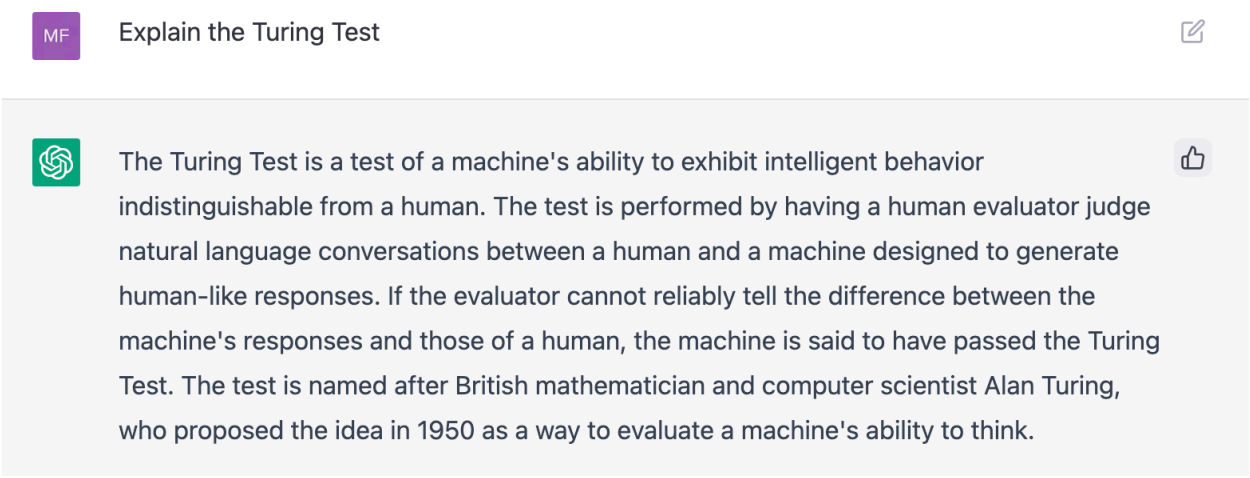


Figure 1. Explanation of Turing Test.

2.5 Machine Learning Chit-Chat Bots

ML systems can learn to chit-chat. They do need data of conversations, and a large quantity of it. There are databases, or datasets, of conversations (e.g. of telephone conversations). Then dialogs from movies are a good source. Also, sometimes, crowdsourcing is used (people either volunteer or are paid to converse in an interesting and potentially useful way). Roughly speaking,

the systems can either retrieve or generate. Retrieval systems will look through the conversations that they are aware of and select fragments similar to the one they are addressing, make a choice or ranking among the fragments, then return a suitable response. Generative systems can 'generate' a response. They learn from their training (and subsequent conversations) what might be suitable in which circumstances. In a sense, they are learning rules. But there are not a fixed number of rules programmed in before the system starts learning. The responses that generative systems produce can be novel or new.

2.6 LaMDA

LaMDA, from Google, is a family of Language Models for Dialog Applications (Thoppilan et al. 2022). These models are for Chit-chat bots. We will briefly describe here what they were designed for and what they can do, and in Chapter 3 we will go into a little more technical detail.

LaMDA agents should be able to supply conversational quality in dialogs lasting 20 turns or more. What they say should be safe. It should be safe, period. Some earlier non-LaMDA dialog agents have suggested that their conversational partners eat ground glass as a treatment for a medical condition. This is not a good or desirable outcome. Additionally, the dialog agent should not offer any views that are biased or toxic. Finally, the conversational contribution needs to be grounded. One might take grounded to mean that the views expressed be true and supported by evidence. But that condition is too strong in as much as there are many areas where there are competing rival views and where all the views have

some reasonableness to them. Grounded is just the requirement that if the agent professes opinions offered as factual then those opinions need not to be fictional or hallucinatory. (Historically, conversational agents have been known occasionally to offer plausible assertions which are pure fantasy.)

2.7 ChatGPT

ChatGPT was produced by OpenAI. ChatGPT itself tells us about OpenAI:

> OpenAI is a research institute that was founded in 2015 by a group of entrepreneurs, researchers, and investors who were interested in advancing the field of artificial intelligence. The organization was created with the goal of promoting and developing friendly AI, which is AI that is aligned with human values and capable of working towards the benefit of humanity. Since its founding, OpenAI has become one of the leading organizations in the field of AI research, and has made significant contributions to the development of ML algorithms and technologies. Some of the notable achievements of OpenAI include the development of the GPT-3 language model, which is one of the most advanced natural language processing systems in the world, and the creation of the DALL-E artificial intelligence system, which is capable of generating images from text descriptions. Overall, OpenAI has played a major role in advancing the field of AI and in fostering the development of safe and beneficial AI technologies (ChatGPT output to the question 'What is the history of OpenAI?' 12/15/2022).

ChatGPT is a Large Language Model (LLM), or a Foundation Model. Quite what those features are will be explained in the next chapters. The LLM chatbots are so much in advance of all the other earlier chatbots that the earlier ones are mostly rendered obsolete and of historical interest only. As of November 2022, ChatGPT was a chat front end to GPT-3.5 (OpenAI

2022a). GPT-3.5 itself is a Generative, Pre-trained, Transformer, iteration 3.5. 'Generative' here means that it can predict, or generate, new output. 'Pre-trained' means that it has a two-stage training. First a pre-training, and then a specific shaping to the final product. Finally, 'Transformer' is the type of machine learning system. GPT-3.5 was current as of December 2022. GPT-4.0 was released in March 2023 (and likely new versions will be arriving).

Progress, or change, in the research area of LLM chatbots, is extremely rapid. Anything written here right now will be out of date in days or weeks. Please be aware of that. We know what LLMs are, how they are produced, and how they work, to a degree. That can be described. There is another caution besides the changing landscape. The large language models are so complex that there is no real description, explanation, or analysis of the details of their internal workings. Likely this will not change anytime soon.

ChatGPT (using GPT 3.5) can:

- Write high school to college level essays, to about an A standard. The written English is definitely of A standard, and usually the factual content (if that is required or appropriate) is also of A quality. This means that it can write essays for students, articles for journalists, and sometimes, or almost, write research papers for researchers.
- Tell jokes, write poetry, and produce cooking recipes.
- Write, or correct computer programs. This also is carried out to a high standard. This makes it a valuable programming, or debugging, tool. (OpenAI and Github had earlier developed Copilot, which is a

> programmer's assistant (Github 2022). Copilot's abilities seem to be built in to ChatGPT).

As part of chatting ChatGPT can:

> … answer follow-up questions, admit its mistakes, challenge incorrect premises, and reject inappropriate requests (OpenAI 2022a).

It does have some weaknesses (which we will look at further in the next Chapter). But briefly:

- It is not good on basic mathematics and arithmetic.
- Its knowledge is not current. (Its training data is from pre-2020. This means that it has no knowledge, for example, of recent politics or sporting events.)
- Sometimes it can go completely wrong. For example, some Users have reported of occasions when ChatGPT has referred to books that do not exist.

The weaknesses that it has have been fully acknowledged by OpenAI (and many of them will be addressed in short order).

Nevertheless, just what it can do is astonishing. It can produce, or generate, grammatical correct and sophisticated English that in most cases could not be distinguished from that written by an educated native speaker. It also has, or appears to have, extensive factual knowledge. There is an example

above (Section 2.4 on the Turing Test). There are a plethora of examples of ChatGPT output available online. (There were one million registered Users within a couple of days of its research release on 11/30/2022, and almost all of those published their experiences on social media, discussion sites (like Reddit), or in online journals or newspapers.) This research release does not reveal everything about its capabilities. For example, it seems to have a problem with truth. It writes plausible English but seems to do that without knowing whether what it is saying is true or has evidence in its favor. Contrast this with the Google Search engine. If you asked that about the Turing test, it would return a ranked list of links. ChatGPT seems much better here, just from a User interface point of view. But if you wanted to re-assure yourself, when using Google Search, you could follow some or all of the links and get a sense of the foundations or grounding of what the contents of the links were asserting. With ChatGPT, truth, fiction, and well-formed writing, seem all to run into one. On tricky topics, or more complicated concepts, ChatGPT sometimes gave highly plausible answers that are flat-out wrong — something its creators warn about in their disclaimers. It may be, so-to-speak, that ChatGPT knows why it says what it does, but we are not fully aware of that at this point. It has been said '*fiction* is what ChatGPT is really good at'.

2.8 Task-Oriented

Most task-oriented dialog agents (including Siri, Cortana, Alexa, and OK Google) use Genial Understander System (GUS) architecture (Bobrow et al. 1977; Daniel Jurafsky and Martin 2021).

> GUS (Genial Understander System) is intended to engage a sympathetic and highly cooperative human in an English dialog, directed towards a specific goal within a very restricted domain of discourse....
> There is good reason for restricting the domain of discourse for a computer system which is to engage in an English dialog. Specializing the subject matter that the system can talk about permits it to achieve some measure of realism without encompassing all the possibilities of human knowledge or of the English language. It also provides the user with specific motivation for participating in the conversation, thus narrowing the range of expectations that GUS must have about the user's purposes. A system restricted in this way will be more able to guide the conversation within the boundaries of its competence. (Bobrow et al. 1977)

This GUS architecture is not itself a ML architecture, rather it is an older AI frame-and-slot design. The implementations do use natural language, English, say, and so the modern dialog agents will use recent NLP machine learned techniques.

A frame is a mini-world, or situation, or ontology, that captures everything that either is known or needs to be known about a certain set up. For example, consider on online take-out system for a restaurant. There will need to be a menu (with appetizers, entrées, deserts, etc.), a recipient's name address, payment details, and so forth. These constitute the frame (or perhaps even several frames), and the frame has slots (one for the appetizers, one for the credit card number, ...). The slots will have types. For example, the purchaser's name is a string (e.g. 'John Smith') and the credit card number is a number (e.g. 1234 5678 9100 1234). What GUS will aim to do is to be genial while filling all the requisite slots with values of the

required type. In principle, it really is not a lot more difficult than form filling. After all, you can order take-out from a form on a web page. Adding a voice interface makes the process harder.

Different computer programmers may well organize the algorithms for a GUS architecture in different ways. We can sketch here the general approach of the 1977 original. The program will try to retain the dialog initiative as much as possible. There will be an agenda loop of events or tasks to be done. This pattern is standard event-loop (or real time) processing. The program will keep going around the loop until it meets an exit condition. Tasks can be added to, or removed from, the agenda. Tasks can be suspended, to be revisited later. There will be an overall frame with slots, and possible many subsidiary frames with their slots. The event-loop of tasks will aim to fill all the slots that need filling. (Some slots may be able to be left open in certain circumstances.) Many of the slots will have their own sub-programs, which will be invoked when the slot is filled. (For example, two slots might need the same value (e.g. a departure city in a reservation system). Then each of these slots might have small program to put a value for the departure city into the other slot, if required, when its own value is filled.) Verbal or textual input will be able to be taken at any time (and added to the processing loop). Output (i.e. responses) will be given at the program's direction. The actual processing of the dialog will be where the real difficulties lie. There will be morphological processing (to pick up the words), syntactical processing (to pick up the sentences), and a good amount more to make sense of the vagaries of dialog and to fit the results into the overall state of the system. Also, the filling of the slots will

itself often require interfacing with one or more databases (for example, to look up the departure times of the trains or airplanes).

2.9 GPTs

On 11/6/2023, OpenAI announced their initiative that provides an infrastructure for users to create and use what they call ‘GPTs’ (Altman 2023; OpenAI 2023d). These are customized variants of ChatGPT which can follow instructions in a natural language, access documents outside of their training (such as pdfs and research papers), and use third party services (such as search engines).

GPTs are extremely easy to create. The process being carried out in natural language. Here is one produced in less than 5 minutes:

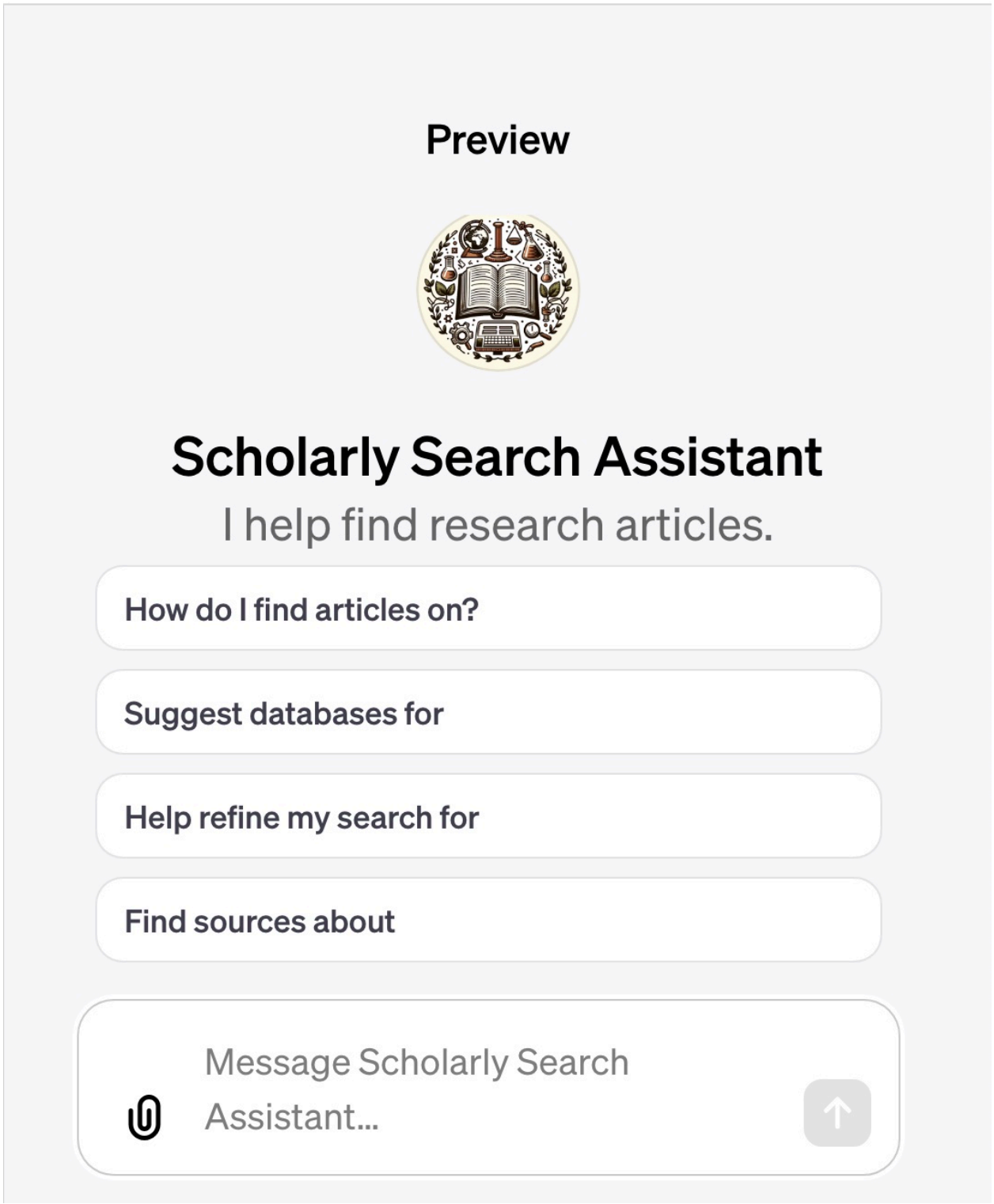


Figure 2. The Scholarly Search Assistant, an Example of a GPT.



Name

Scholarly Search Assistant

Description

I help find research articles.

Instructions

Scholarly Search Assistant will engage users in a conversational tone, making the search for machine learning and librarianship articles more approachable. It will provide guidance and suggestions using everyday language, avoiding jargon unless it's part of the user's query. The GPT will use its own judgment to provide general answers when requests are vague, ensuring users receive helpful information even from broad inquiries.

Knowledge

Upload files

Capabilities

- [x] Web Browsing
- [x] DALL·E Image Generation
- [x] Code Interpreter

Actions

Add actions

Figure 3. The Configuration of the 'Scholarly Search Assistant'.

At this point, with GPTs, the production of many chatbots is nearly trivial. Quite how they work is another question. By the end of Chapter 5 on Large Multimodal Models, we should have some reasonable insights on that.

GPT-4 Turbo, released November 2023, has processing images and voice synthesis built in (i.e. it is multimodal). The GPTs are going to use GPT-4 Turbo as their computing engine. Shortly the chatting will be able to be by text or by audio.

2.10 Annotated Readings for Chapter 2

Altman, Sam, dir. 2023. *OpenAI DevDay, Opening Keynote*. https://www.youtube.com/watch?v=U9mJuUkhUzk. (Altman 2023)

c.ai. "Character.Ai." character.ai. Accessed July 3, 2023. https://beta.character.ai/. (c.ai 2023) This allows you to chat with chatbots that either you or others have created.

Fernandez, Peter. ""Through the Looking Glass: Envisioning New Library Technologies' AI-Text Generators as Explained by ChatGPT." *Library Hi Tech News* 40, no. 3 (2023): 11–14. https://doi.org/10.1108/LHTN-02-2023-0017. (Fernandez 2023)

G2. "Best Bot Platforms Software." G2, 2023. https://www.g2.com/categories/bot-platforms. (G2 2023) . Just scan this.

Jurafsky, Daniel, and James H. Martin. "Chapter24 Chatbots & Dialogue Systems." In *Speech and Language Processing*. Daniel Jurafsky & James H. Martin. Draft of December 29, 2021., 2021. https://web.stanford.edu/~jurafsky/slp3/24.pdf. This is the standard text. It is pitched at an intellectual level in advance of that which we are using here.

Library Hi Tech News. "Special Issue on ChatGPT." *Library Hi Tech News*. 40, no. 3 (2023). This has a number of useful articles on the uses of ChatGPT, especially in a library setting. (Library Hi Tech News 2023)

OpenAI. “Introducing GPTs,” 2023. https://openai.com/blog/introducing-gpts. (OpenAI 2023d) You can try creating GPTs yourself at chat.openai.com/create.

Schlicht, Matt, and Ben Parr. “*Chatbots Magazine: The #1 place to learn about chatbots*.” Chatbots Magazine, 2023. https://chatbotsmagazine.com/. This is a good resource, it is a collection of short articles by a variety of authors. (Schlicht and Parr 2023)

Weizenbaum, Joseph. “ELIZA—a Computer Program for the Study of Natural Language Communication between Man and Machine.” *Communications of the ACM* 9, no. 1 (1966): 36–45. https://doi.org/10.1145/365153.365168. This is an important original paper. It is very readable.

Wolfe, Matt. “Future Tools - Find The Exact AI Tool For Your Needs,” 2023. https://www.futuretools.io/. (Wolfe 2023). This is an astonishing and valuable resource. Especially recommended are the News and Videos sections. The content is not really what one might call ‘academic’. Rather it focusses on what the commercial companies and start-ups are producing. The content slightly favors work with images and video, whereas our interest is a little more with text and information.

Chapter 3: Language Models

3.1 Introduction

Many of the modern AI systems have their origins with language models. These typically use text as input and produce text as output. Language models have a huge advantage, when considered as a ML research challenge. They can be trained on (unlabeled) text, and there is plenty of that (for example, the Internet). Elsewhere, any ML program, trained by supervised learning, will need a quantity of high-quality labeled data, and that is hard to come by. But most language models will be able to be trained by self-supervision using digitized text.

Emily Bender and her co-authors explain language models as:

> … the term language model (LM) … [refers] to systems which are trained on string prediction tasks: that is, predicting the likelihood of a token (character, word or string) given either its preceding context or (in bidirectional and masked LMs) its surrounding context. Such systems are unsupervised and when deployed, take a text as input, commonly outputting scores or string predictions. (Bender et al. 2021)

What does this mean? Some background theory will be useful here.

3.2 Markov Chains

Alexander Pushkin, the eminent Russian poet, playwright, and novelist, writes:

> My uncle's goodness is extreme,
> If seriously he hath disease;
> He hath acquired the world's esteem
> And nothing more important sees;
> A paragon of virtue he!
> But what a nuisance it will be,
> Chained to his bedside night and day
> Without a chance to slip away.
> Ye need dissimulation base
> A dying man with art to soothe,
> Beneath his head the pillow smooth,
> And physic bring with mournful face,
> To sigh and meditate alone:
> When will the devil take his own! (Pushkin 1881)

This is from an English translation of *Eugene Onegin: A Romance of Russian Life in Verse* written in Russian.

There is an interesting tidbit of information associated with this. Until about 1900, probabilistic analyses of common sequences of apparently random events assumed that the individual events were independent one from another. So, as examples, that the second and subsequent throws of a single gambling die are not influenced in any way by the earlier throws, and that the outcome of a spin of a roulette wheel does not depend on the spin, or spins, that went before. The Russian mathematician Andrei Andreevich Markov questioned whether this independence assumption held of all

sequences, especially in fields outside of gambling. He conjectured that it did not and produced a suitable mathematical theory to cover the case.

The basic theory is relatively easy to understand. Almost everybody is familiar with the centuries old board game Snakes and Ladders. In this, there is a board with a hundred and one numbered squares on it, and players advance from the start (i.e. 0) to 100 by repeatedly throwing dice and using the numbers they obtain. The first player to 100 wins. On the board there are some 'snakes' and some 'ladders'. These connect pairs of numbers, usually separated by 5 to 20 intervening numbers. The heads of the individual snakes are higher— of a higher number— than the tails. If a player lands on the head of a snake, they have to go backwards to the tails, losing some of their advance. If a player lands on the base of a ladder, they advance to the top of that ladder. Consider just one player, and one die, and assume the player throws a 5. This may advance the player 5 spaces, but it also may send the player back some spaces or it may send the player forward more spaces than 5. What happens depends in part on what square the player is on. Throwing a 5 when on square 15 might produce an entirely different result, in terms of going forward or backward, than throwing a 5 when on square 32. The square that the player is on is the *state*. The throw of the single die will produce one of six numbers. This will move the player to one of six other states. The values of the throws (in this case) are equally probable, and so there are six equally probable non-zero *transition probabilities* from the current state to the next state. What happens with a throw depends only on the current state (and the value of the throw). It does not depend on the state before the current state, or any before that, i.e. on the history of the individual game. For example, if a player is on square

27, state 27, that is all that matters— it does not matter how the player got to 27. Snakes and Ladders play is an example of a Markov Chain. Conceptually, there is usually time, or a time step, or time beats, involved in producing the sequence of states in a Markov Chain. If we wanted to build that into the example, we could just require that the player throws the die every 30 seconds. There are a few other details that can be added to the Snakes and Ladders example to make it closer to Markov Chain theory. There is usually a start state, or a probability among the states for one being the start state. So, the zero square would have a probability of 1 of being the start state (and the other states 1-100 probability zero). Similarly, there might be a terminal state. If so, there would be no transition probabilities for a move out of the 100 square. Also, it is not possible to rest on the foot of a ladder or the head of a snake (because the player would be required to move on elsewhere). This could be represented either by having no transition probabilities that end with the foot of a ladder, or the head of a snake, and including the jumps in with the dice values, or just by omitting those 'states' altogether. Summing up Markov Chains, there is a sequence of events, states, transition probabilities, and the dependence on the state is just with the current state. This latter is the key feature of Markov Chains, the dependency is with a single state only, the current state, and not with earlier states. (Sometimes this is called the Markov Property.)

What did Markov do with *Eugene Onegin?* He divided the first 20,000 characters of the text into consonants and vowels (in Russian, of course). This gave him a value for the *overall probability* of a particular character being a consonant (or being a vowel). He then looked at pairs of consecutive characters and what the probabilities were for the *second*

character being a consonant (or a vowel) if the first character was a vowel (or a consonant). These are transition probabilities from vowels to consonants, and consonants to vowels. What he found was that vowels were more likely to be followed by consonants (and consonants by vowels)— that the transition probabilities were different to the overall probabilities. What this meant was that if the text was thought of as a sequence or stream of letters, the appearance of one letter after another was not an independent event, being governed by only the overall probabilities of the letters, rather it was a dependent event having a dependence on the current letter or state. From here, there is the theory and mathematics of Markov Chains.

Markov Chains are everywhere. They are in text. Not just with consonants and vowels, but also with letter sequences like 'u' following 'q', and word sequences like the word 'He' being followed by a verb phrase. They are in spoken text, or speech, with sequences of phonemes (certain sounds follow others). They are in the weather (rainy days tend to follow other rainy days). Many games— the sequences of their moves— can be analyzed by Markov Chains (e.g. chess). The Stock Market can be seen as a Markov Chain (bull markets tend to be followed by bull markets, and bear markets followed by bear markets). There are many Markov Chains in biology, for example with DNA sequences. More-or-less any values or measurements that change over time, such as prices in a stock market, speeds of motor cars, etc., can be modeled by Markov Chains. [There is a paper *Five greatest applications of Markov Chains* (Von Hilgers and Langville 2006). It is recommended.]

3.3 Hidden Markov Models

Hidden Markov Models (HMMs) are extremely common in the ML analysis of sequential data that has a dependence on time. Typically, a model will be able to explain a sequence and predict or generate new sequences. The techniques are most at home with Natural Language Processing (NLP) and with handwriting recognition, speech recognition, and biological informatics.

With an HMM there are two collections of probabilities. In the background, there is a probabilistic Markov Chain or Process as explained above. But some, or all, of the states of this Markov Process are hidden (and unseeable directly by observers). However, the hidden states 'emit' observations which can be seen or observed. These emissions of observations are themselves governed by probabilities relating them to the hidden states. An example might help with understanding the set-up. Suppose we are interested in the sequences of year-on-year climate values in New Zealand from a thousand years ago. (We may be interested in this because the climate affects the environment, its flora and fauna, and their history and development.) Suppose the years can be just hot or cold, and which they are for a particular year depends solely on whether the predecessor year is hot (or is cold). Suppose also that the probabilities of a hot year being followed by another hot year (and a cold year being followed by a cold year) are all known from modern values (which are assumed to be unchanged from those of a thousand years ago). These ancient weather sequences are Markov Processes, and their states are hidden and not directly observable. However, the weather affects tree-ring growth, and we have modern

probabilities relating that growth to the weather (which, again, are assumed to be unchanged from a thousand years ago). More than a few New Zealand Kauri trees live longer than a thousand years. So, the Kauri rings can serve as modern emitted observations on the Hidden Markov Process of ancient weather sequences in New Zealand (Stamp 2017; Rabiner 1989).

We will not address the mathematics in detail here. But what the mathematics can do is i) estimate the probabilities for a hidden sequence, give the observable sequence, and the other relevant probabilities ii) estimate the probabilities for an observable sequence, given the hidden sequence, and the other relevant probabilities, iii) given an observable sequence, the number of hidden model states and different types of observations, can use successive approximation on the parameters to produce values which give the greatest probability of yielding the provided sequence. (The successive approximation uses what is known as 'gradient descent', see the excellent videos from Josh Starmer or Grant Sanderson for explanations of gradient descent (Starmer and StatQuest 2019; Sanderson and 3Blue1Brown 2017b). Gradient descent is a standard ML technique.)

As mentioned, Natural Language Processing often uses HMMs. Mark Stamp, in his (Stamp 2017), gives a simple example of language analysis, which is more-or-less the inverse of Markov's *Eugene Onegin* case. Say you gave 'Marvin the Martian' (an alien) a corpus of a million words of actual English text and invited him/her/them to investigate sets of individual characters under the assumption that there was a HMM, with two hidden states, that was producing the observed English. The result of Marvin's research would be that there were two relevant sets of letters, and they

would be the vowels and the consonants. Initially, Marvin knows nothing about English, yet Marvin can learn structural and statistical properties using an HMM.

Marvin, and Markov, in the *Onegin* example, are looking at individual characters or letters. But what is much more useful and widespread in the general NLP case is analyzing entire words, or parts of words, and, for example, tagging them with their parts of speech (POS). In this, HMMs are often used to identify nouns, verbs, adjectives, determiners, etc. Parts of speech point to the function of the words in the sentences i.e. they hint at role or meaning.

3.4 Shannon's Guessing Game

3.4.1 Introduction

An interesting precursor of modern NLP research is the 1940s work of Claude Shannon on signal information (C E Shannon 1948). Shannon provided an extensive analysis of the natural language English and it is worthwhile to follow and develop his reasoning. He was trying to calculate how much entropy (signal information) there was in passages of English. (What entropy and signal information are need not concern us.) The approach here was to use successive approximation by models. Shannon's zero-order model assumed that the alphabetic letters or characters had an equal probability of occurring in English text. But in real English, assuming here that the speaker or writer is not idiosyncratic, the occurrence of letters is not equiprobable; 'e', for example, has a probability of 0.13 of occurring and 'w' a probability of 0.02. If the differing probabilities for all the letters

are taken into account, that produces what is known as the first-order model. There are also conditional, or transition, probabilities relating letters in letter sequences in English; 'q', for example, is almost invariably followed by 'u'. This model is now using a Markov process for analysis. Two letter sequences, giving the second-order model, can be considered, or three letter sequences and so on. A similar approach can be adopted to predicting a missing word or the next word, using whole words as the alphabet and not just single letters. (Filling in the blanks in some text is known as a 'cloze task'. These will be explained in more detail shortly).

The assumption of non-idiosyncrasy of the source English text is entirely in order— John Pierce tells us of E.V. Wright who published the 267 page novel *Gadsby* in which there are no 'e's (Pierce 1980) p.48). There is an important point to be made here. When there is focus on the probabilities of occurrences of letters in text, say 'e's, there is a need to address context. Such probabilities may well have different values depending on, for example, whether the text is *Gadsby*, the *Sunday Times*, or every piece of English on the Internet. Care is also needed here with the relation between the training data and the intended real use data (the test data) for ML language probabilities. You would not want to use Gadsby as the sole training data for all of written English.

Shannon conducted extensive experiments on English. One technique he used was to give test subjects many letters of a sequence of English—usually up to about 50 letters—and then ask them to guess the next unknown letter (a cloze task); he found that the subjects were able to guess correctly about half the time or more. (You can try Shannon's experiment or game :- pick

roughly the middle letter of any line in this book, count forward fifty letters, keeping the fifty first letter covered before making your guess, and see how uncertain you are about the value of the fifty first letter.)

The results show that English is massively redundant. This redundancy has its advantages. Natural languages are highly noise tolerant— often many letters in a message can go missing or be mis-transcribed and yet still the receiver will be able reconstruct the original. (And also, yet again as Shannon pointed out, without redundancy, crossword puzzles would be impossible.)

English is usually taken to be an ergodic source (roughly, that means that one reasonable passage is a good representative of the whole, as far as all the probabilities are concerned). However, so-called Large Language Models and Foundation Models would typically use vast amounts of text as input data (see, for example, Common Crawl (Common Crawl 2022)). There is a good reason for that. As we will see these models are trying to do more than evaluate probabilities; they are, for example, also trying to acquire knowledge (such as, the fact that George Orwell is the author of the book *1984*).

3.4.2 Shannon's Approximations as Markov Processes

[An indicative fact: Shannon's seminal 1948 paper originally had the title *A Mathematical Theory of Communication*. In 1949, when reprinted in

(Claude Elwood Shannon and Weaver 1949), it was renamed *The Mathematical Theory of Communication*. Enough said.]

To explain the GPT Large Language Models, there is a passage in the Shannon paper which is pure gold:

> To give a visual idea of how this series of processes approaches a language, typical sequences in the approximations to English have been constructed and are given below. In all cases we have assumed a 27-symbol "alphabet," the 26 letters and a space.
>
> 1. Zero-order approximation (symbols independent and equiprobable).
>
> XFOML RXKHRJFFJUJ ZLPWCFWKCYJ FFJEYVKCQSGHYD QPAAMKBZAACIBZL- HJQD.
>
> 2. First-order approximation (symbols independent but with frequencies of English text).
>
> OCRO HLI RGWR NMIELWIS EU LL NBNESEBYA TH EEI ALHENHTTPA OOBTTVA NAH BRL.
>
> 3. Second-order approximation (digram structure as in English).
>
> ON IE ANTSOUTINYS ARE T INCTORE ST BE S DEAMY ACHIN D ILONASIVE TU- COOWE AT TEASONARE FUSO TIZIN ANDY TOBE SEACE CTISBE.
>
> 4. Third-order approximation (trigram structure as in English).
>
> IN NO IST LAT WHEY CRATICT FROURE BIRS GROCID PONDENOME OF DEMONS- TURES OF THE REPTAGIN IS REGOACTIONA OF CRE.
>
> 5. First-order word approximation. Rather than continue with tetragram, ..., *n*-gram structure it is easier and better to jump at this point to word units. Here words are chosen independently but with their appropriate frequencies.
>
> REPRESENTING AND SPEEDILY IS AN GOOD APT OR COME CAN DIFFERENT NATURAL HERE HE THE A IN CAME THE TO OF TO EXPERT GRAY COME TO FURNISHES THE LINE MESSAGE HAD BE THESE.
>
> 6. Second-order word approximation. The word transition probabilities are correct but no further structure is included.
>
> THE HEAD AND IN FRONTAL ATTACK ON AN ENGLISH WRITER THAT THE CHARACTER OF THIS POINT IS

> THEREFORE ANOTHER METHOD FOR THE LETTERS THAT THE TIME OF WHO EVER TOLD THE PROBLEM FOR AN UNEXPECTED.
> The resemblance to ordinary English text increases quite noticeably at each of the above steps. Note that these samples have reasonably good structure out to about twice the range that is taken into account in their construction. Thus in (3) the statistical process insures reasonable text for two-letter sequences, but four-letter sequences from the sample can usually be fitted into good sentences. In (6) sequences of four or more words can easily be placed in sentences without unusual or strained constructions. (C E Shannon 1948, 7)

Shannon starts here with equiprobable letters, the zero-order approximation, then moves to allow the letters to have the probabilities that they do in real English. Then, with the second-order approximation, he allows the letters to have transition probabilities— probabilities dependent on the single preceding letter only. So, if the preceding letter were 'Q', for example, the probability of the next letter being 'Z' would be zero, the probability of it being 'U' would be high, the probability of the next letter being a blank space would have some positive value, and so on. At this point what is going on is a Markov process. Shannon tells us this:

> Stochastic processes of the type described above are known mathematically as discrete Markoff processes and have been extensively studied in the literature. (C E Shannon 1948, 8)

The preceding letter in the second-order approximation is a *state* or *context* and the relevant probability depends on that state. The third-order approximation uses transition probabilities, but this time extends the state or context to the two preceding letters. At this point, Shannon switches to using entire words, not individual letters, and that is not of further interest to us here. A point to note is that 1948, Shannon would have had no way of

calculating transition probabilities using three, four, five, ..., and so on preceding letters. (Basically, you need computers and digitized text to do that.) Had he been able to do so, no doubt he would have explored much larger contexts.

3.4.3 Training a Shannon-Markov Model to Produce 'A Baby GPT'

Nowadays, we can train a Shannon-Markov model to evaluate the transition probabilities for itself, using self-supervised learning. The result is what is in effect a GPT (a Generative Pre-Trained Transformer). Andrej Karpathy suggested the following as an example (Karpathy 2023a). There are some serious simplifications in this.

> The alphabet consists of two tokens {A,B}
>
> The context (or prompt) is the three preceding letters. So, there are eight states (namely AAA, AAB, ABA, ABB, etc.) There will be 16 transition probabilities from the contexts to the next token, namely
>
> > AAA->A with probability unknown
> > AAA->B with probability unknown
> > AAB->A with probability unknown
> > AAB->A with probability unknown
> > Etc.
>
> These probabilities need to be determined from the training data, by training. [We can note, though, each context can produce either A or B and if the probability for one of those is x the probability

for the other is (1-x). So, there are eight values to be pinned down.]

The training data, the training text, is the sequence of 15 tokens:

AAAABAAAABAAAAB

What our ML system is going to do is to produce a 'next token' as an output, then do that again, and again, ..., to give 15 tokens. It needs a 3 token context to start with. So, we will give it three As as its initialization or prompt

AAA

then let it run to produce the next 12 tokens. Just to explain further. The first prompt will produce an output token, say B.

AAAB

Then, the new token gets included in the next prompt and first token in the earlier prompt gets dropped out of the prompt, and, A, say, is generated.

AAABA

So there is a continuous 'adding, and dropping, and sliding forward'. The ML system sees only three symbols at a time, the ones shown in green, but those three symbols change through time.

Suppose, for the first training round, it completes its generation and produces

AAABBAABABAABAB

This output will be compared with the training data. There will be a scoring system that evaluates it. Then the machine learning model will revise its internal weights (i.e. parameters). It will

improve itself. And there will be a second training round, and a third, and the process will be repeated over and over. Karpathy has provided sample software to do this, and typically there might be 50 rounds of training. In his publication, the training determined the transition probabilities to be

AAA->A with probability 0.45
AAA->B with probability 0.55
AAB->A with probability 0.78
AAB->A with probability 0.22
Etc.

Looking from outside the system we can see what the transition probabilities should be. The context is the three previous tokens. We can slide the context window through the training text seeing what the outcome tokens are.

AAAABAAAABAAAAB *
AAAABAAAABAAAAB *
AAAABAAAABAAAAB
AAAABAAAABAAAAB !
AAAABAAAABAAAAB
AAAABAAAABAAAAB *
AAAABAAAABAAAAB *
AAAABAAAABAAAAB
AAAABAAAABAAAAB !
AAAABAAAABAAAAB
AAAABAAAABAAAAB *
AAAABAAAABAAAAB *

To take two examples from this, the context AAA appears 6 times (an asterisk picks them). Three of these are followed by A, and three followed by B. So, roughly speaking, we might expect that both:

AAA->A
AAA->B

would have a probability of near 0.5 (in fact, the published training run gave 0.45 for one and 0.55 for the other). For the second example, the context ABA appears twice, identified by an exclamation mark, and each time it is followed by an A. So, ABA->A should be 1.0 and ABA->B should be 0.0 (the published training run gave 0.78 and 0.22). There were only 50 rounds of training. If more rounds were used, the values would sharpen to the correct values.

Notice that some of the possible contexts do not appear in the training data. For example, BBB does not. So, what should happen with a BBB context is an open question. It is not determined by the training data. It is at this point that what is called *inductive bias* enters. The system needs to make a 'reasonable' guess and with the Karpathy sample implementation it goes with 0.5 chance of getting an A, and 0.5 of getting a B, for all contexts it cannot determine from the data.

At this point, we have a 'baby GPT'. It is a Shannon/Markov model with a context of 3 preceding tokens. It can generate output. Here is one example that it produced:

AAABAAABAAABAABAAAABABA

To compare the baby with a GPT like GPT-3. Our alphabet has 2 tokens, a full-blown GPT would likely use 'words' and not individual letters in its alphabet and it might have an alphabet of 50,000 different tokens (words). Our context is a context of the 3 previous tokens, a full-blown GPT might use a context window that can see the 30,000 previous tokens. As of May 2023, Anthropic's Claude model has a context window of 100,000 tokens

(which is about 70,000 words— its tokens are larger than single characters, but smaller than entire words). Our training text is 15 tokens long. GPT-4's training text is more-or-less 'the entire Internet'. We can train in milliseconds. GPT-4 takes months on parallel 'supercomputers'. The baby was trained on 50 rounds ('epochs') of training. OpenAI has not disclosed how many epochs of training it used (we asked GPT-4 this question) but it will have been in thousands.

3.5 Taylor's Cloze Procedure

In 1953, Wilson L. Taylor published the paper *"Cloze Procedure": A New Tool For Measuring Readability* (W. L. Taylor 1953). Basically, cloze procedures amount to taking some text and deleting parts of it, then inviting the human subjects (or software) to fill in the blanks to recreate the original text. Taylor writes:

> Given "Chickens cackle and --- quack," almost anyone can instantly supply "ducks." .....
>
> > Note that the sentence pattern is a complex one made up of many sub-patterns. One must know not only the meanings (i.e., patterns of symbol-meaning relationships) and forms (patterns of letters) of all the five words, but also the meanings of given combinations of them— plus the fact that the sentence structure seems to demand a term parallel to "cackle" but associated with ducks instead of chickens. In other words, one must guess what the mutilated sentence means as a whole, then complete its pattern to fit that whole meaning. (W. L. Taylor 1953)

As the title of his paper indicates, Taylor's original idea was to use cloze procedures as one tool to assess the reading skills of subjects. But

subsequently their use has expanded far beyond this. Cloze tasks, or cloze procedures are reasonably demanding and they are a popular research technique for modern Natural Language Processing (NLP). [Nowadays, it is a common practice for researchers to write '[Mask]' to show where the blanks are. So, Taylor's illustrative sentence would be written:

> Given "Chickens cackle and [Mask] quack," almost anyone can instantly supply "ducks."

]

3.6 nanoGPT and an Illustration of Training

Andrej Karpathy has provided for us nanoGPT (Karpathy [2022] 2023). This is some software you can run at home (do try it!). It is:

> The simplest, fastest repository for training/finetuning medium-sized GPTs. [It can reproduce] GPT-2 (Karpathy [2022] 2023)

nanoGPT is a language model, and it can be used to illustrate training. Aatish Bhatia has done exactly that in the implementation he calls BabyGPT (Bhatia 2023). One of the examples that Bhatia provides concerns Jane Austen. To paraphrase Bhatia, BabyGPT is provided with the 800,000 words of Jane Austen's work and also a prompt:

> "You must decide for yourself", said Elizabeth [We assume here that this prompt is not among the actual work of Austen, otherwise that would ruin the training, inviting what is called overfitting (here this means that it has been told an answer in its training).]

And what BabyGPT has to do, its task, is to produce the next seventy or so words written in the style of Jane Austen. The process will be for it to output about 700 characters, one at a time. It will produce one character. Then having done that, produce the second character, and so on, until it reaches about 700 characters. A blank space is a character (as are punctuation marks and upper-case characters). Thus, some character sequences will have the appearance of sequences of words (or even of phrases and sentences). There will be a scoring system that evaluates the mistakes. Learning will be used to change the parameters of BabyGPT. Then this will be done again, and again, forming round after round. In the first round it produced:

> "You must decide for yourself", said Elizabeth
> grThbE22]i10anZOj1A2u'T- t'wMOZeVsa.f0JC1hpndrsR
> 6?to8j7dCVCyHwrWFYYGr"X8,IOwC!WAE_]!LtZf8&Or6d'KDiD
> 77Wq'Y4NtV:_'N [and more]

Here, it is producing characters randomly, of equal probability. This is Shannon's zero-order model. Were this first round to be run again, with the parameters re-set to their original values, the output would likely be different. [A character set, with special characters in it, might have a repertoire of 256 different characters. So, the probability for any particular character appearing in any particular position is 1/256. Getting the same output sequence twice is very unlikely.]

After 250 rounds of training, it produced:

> "You must decide for yourself", said Elizabeth gra buseriteand the utharth s atouchanders nd shadeto the se owrerer o thino athe athetlf w wad asire [and more]

This is now similar to Shannon's first-order model. It has figured out that not all letters are equiprobable (for example, that special characters do not occur very often and that 'e's occur relatively frequently). After 30,000 rounds it produced:

> "You must decide for yourself", said Elizabeth rather repeatedly; "that is very agreeable displeasure, they will ever be a lively young woman as it will be more disagreeable." "My dear Fanny, [and more]

Now there are words, sentences, some grammar, and a character stream approximating English. This training was carried out on a laptop computer in an hour. Some of the large language models, which will get to shortly, might use what effectively is a super-computer for months for their training. Also, they may use a two-stage training. There might be a pre-training, which would be a self-learning training similar to the above. This then might be followed by reinforcement learning where there is a panel of about forty human judges giving feedback and a scoring as to the quality of the output.

3.7 Embeddings

> In machine learning, an embedding is a way of representing data as points in n-dimensional space so that similar data points cluster together (Markowitz 2022).

A point in an n-dimensional space can be thought of as a list, or vector, of numbers. As examples, a point in a 2-dimensional space might be [2,3]; a point in a 3-dimensional space might be [1,29,2]; a point in a 7-dimensional space might be [29.1,-7,2.9,13,21.6,-37.23,9]; you get the idea. Then a measure can be put on these lists that will show whether two or more lists are 'similar'. If they are, this means, or might mean, that the underlying data, the relevant underlying data points, are similar.

Embeddings are important.

> ...embeddings power:
>
> 1. Recommendation systems (i.e. Netflix-style if-you-like-these-movies-you'll- like-this-one-too)
> 2. All kinds of search
>   a. Text search (like Google Search)
>   b. Image search (like Google Reverse Image Search)
> 3. Chatbots and question-answering systems
> 4. Data preprocessing (preparing data to be fed into a machine learning model)
> 5. One-shot/zero-shot learning (i.e. machine learning models that learn from almost no training data)
> 6. Fraud detection/outlier detection
> 7. Typo detection and all manners of "fuzzy matching"
> 8. Detecting when ML models go stale (drift)
> 9. So much more!
>
> (Markowitz 2022)

We will get to most of these applications later.

A good example of a typical use of embeddings is that provided by Cohere with its embeddings of Wikipedia articles (Reimers and Alammar 2023; Kayid and Reimers 2022). What this does is to break millions of Wikipedia articles down into several millions of passages of text. Then each passage is

assigned a vector (i.e. a list) of numbers. Conceptually, these vectors are collected together into a vector database (i.e. a library of vectors). Then, as a typical use, were a user to search for a topic, or a phrase, or a query, that search phrase would be embedded to produce its own vector. Finally, the vector database would be searched for vectors similar to the search vector and the relevant actual Wikipedia passages would be retrieved and returned. There is an initial embedding process to produce vector database, another embedding process to embed the search phrase, then a 'de-embedding' process to get the sought for Wikipedia texts back from the matched similar vectors.

Suitable embeddings of text will embed meaning, not the surface form of the words, phrases, and sentences. One consequence of this is that the application or software can work with multiple languages:

> … rely[ing] on the property that sentences that are similar in meaning will have similar embeddings, even if they are in different languages. (Reimers and Alammar 2023)

One extremely important point about embeddings is that there is not just one way of producing an embedding. There are indefinitely many ways of doing it, usually each with their own algorithms or software. This matters because the embedding algorithm used to produce the vector database and the embedding algorithm used to embed the search phrase need to be the same. Care is needed here because the database will be produced at one point in time, say 2021, yet people might still be searching it 5 or 10 years later. Typically, developers outside of the largest companies will not create embedding algorithms for themselves. Rather they will use algorithms from

OpenAI, Hugging Face, or other open-sourced or licensed resources. This exposes the developers to possibly catastrophic risk. If the algorithms, or their licenses or availability, are changed, the developers' software might be irretrievably broken. To give a concrete example, later in the text we use OpenAI's text-embedding-ada-002 algorithm to embed 'Stochastic Psittacosis'— were OpenAI to withdraw its provision of ada-002 that result basically would become garbage. This matters not just to developers. Purchasers of software using these technologies need to be wary of becoming dependent and 'locked in'. The risk here is real. In November 2023, OpenAI was close to imploding as a company (with the four-day sacking and re-instatement of its CEO, Sam Altman).

3.8 Word Embeddings and Word2Vec

Word embeddings are a common and useful tool or technique to process words prior to their being used in ML models. Often the embeddings can capture useful deep syntactic and semantic information. The DL Models themselves use numbers internally for their arithmetic and mathematics. This gives the preliminary task of converting words to numbers for use in the models. This coding could be done trivially by using the number 1 for the first word, 2 for the second word, and so on. But notice this. Say word number 3 was 'toy' and word number 47 was 'toys'. Those two words are similar to each (one is the plural form of the other). But information of that similarity is lost in the coding. The numbers 3 and 47 are not similar, nor do they give any hint of similarity between the words that they are coding. Tomas Mikolov, and fellow authors, following earlier work, proposed and developed the technique of using vectors (lists of numbers) to code the

words (Mikolov, Chen, et al. 2013; Mikolov, Sutskever, et al. 2013; Mikolov, Yih, and Zweig 2013; Colyer 2016; Alammar 2019). The idea here is that in a large corpus of text the word 'toys', for example, appears in various contexts (that is, surrounded by other words e.g. 'children', 'Xmas', and so forth) and it tends not to appear in other contexts (e.g. surrounded by words about the fourth quarter of the Super Bowl); then some aspects of these appearance contexts can be placed into the vector that is to represent 'toys'. Similarly for the word 'toy' and its contexts. The overlap of the contexts gives information about the words. This type of coding, exemplified in the software Word2Vec, can capture similarities and also some real information. It can open the way to answering questions like:

Paris is to France as [Mask] is to England

The famous example that Word2Vec is known for is:

King – Man + Woman = Queen

We are using a rough representation here of vectors and vector arithmetic. But what Word2Vec is telling us is: if you take the concept of King, take the Man out of it, add a Woman to it, you get the concept of Queen. Word2Vec has reached this conclusion merely by being trained on a large corpus of text.

Word2Vec is not the only way of producing word embeddings, nor is it the best for all circumstances. But, the point is, the use of intelligent word embeddings on a large corpus of text can provide to ML NLP programs

input that has some syntactic and semantic information built in (see also (Tenney et al. 2022)).

3.9 Adding Knowledge to Language Models

In the case of natural languages, language models, in their original and simpler forms, had to manage words, parts of speech, grammar and so forth. They did not address knowledge or truth or meaning or reference. So, for example, if such a model produced, for some purpose, some entirely correct syntactical and grammatical output such as 'The Mississippi is the longest river in the world and it is 30 miles long' that would be perfectly good and the fact that what is being asserted is mistaken is beside the point. [This has consequences for the training data. The data needs to be good sample English, say, with all its varying genres, styles, and diversity. But it does not need to be vast and encyclopedic in its content. It does not need, for example, to contain simple, if obscure, everyday facts like the name of the Secretary of the Teamsters Local Union LU No 2. (For those with an enquiring mind, that name is 'Erin Foley'.)]

But we have already seen value in many settings to add knowledge to ML systems. For example, a reservation chatbot, a task-oriented dialog agent, linked to a database, can book airline flights and be absolutely correct on the schedules, and prices etc. It does not take a huge stretch of the imagination to realize that many published texts contain knowledge, for example, a favorite children's book Bertha Morris Parker's *The Golden Book of Facts and Figures: A Treasury of Information on Hundreds of Subjects With More than 500 Pictures in Color* does, so do many

encyclopedias, and, indeed, Wikipedia. Most of our knowledge is in books and libraries. Presumably a language model could be configured to take on cloze challenges like:

> The [Mask] river is the longest river in the world.

and answer it with 'Amazon'. Actually, many ML question answering systems can do exactly this. Further, Fabio Petroni, and his fellow authors, inform us that larger language models are developing this ability (Petroni et al. 2019). There is an interesting and important point here. Historically, the way that ML systems would interact and interface with knowledge in text is by producing or extracting from the text a database and then addressing queries to that database. But this is not easy to do, for a variety of reasons. Databases, with their tables of rows and columns, need design. They need ontologies. Different databases may well have different ontologies and be 'information silos' unable to share their knowledge one with another. Also, the training of database extractors would likely need to be *supervised* training (i.e. training that used labeled data), and once labeled data is required a project becomes much more challenging. Then the questions to a database have a structured semi-formal logical structure, and these might not sit easily with Users' expectations. But research now seems to indicate that making the language models larger and larger, and training them, *unsupervised* or *self-supervised*, on more expansive corpuses of text, gives those models question answering, and knowledge manipulation abilities that are superior to the earlier database extraction techniques (Lewis, Denoyer, and Riedel 2019). Of course, research can be revised by further research, and promising research can fade to nothing. But right now, 2023,

it looks as though making the language models larger, and trained more extensively, will yield gold.

3.10 InstructGPT and the Insights it Provides

As of January 2023, language models, and ML in general, have been well and truly being brought to the world by ChatGPT and similar astonishing ML programs (OpenAI 2022a). ChatGPT was created by OpenAI (with a billion dollars or more funding from Microsoft, and more funding from others) (OpenAI 2022c). Google and Facebook, and maybe other commercial companies or institutions, have similar programs (e.g. Bard and LlaMA). OpenAI have not to date published information on how ChatGPT works (and they may not do so in detail— they are a for-profit commercial company). However, they have said that it works the same way as InstructGPT, which is one of their earlier programs, and they have published a thorough research paper on InstructGPT (Ouyang et al. 2022). We should look at InstructGPT.

Long Ouyang and his fellow authors provide the following abstract to their paper:

> Making language models bigger does not inherently make them better at following a user's intent. For example, large language models can generate outputs that are untruthful, toxic, or simply not helpful to the user. In other words, these models are not aligned with their users. In this paper, we show an avenue for aligning language models with user intent on a wide range of tasks by fine-tuning with human feedback. Starting with a set of labeler-written prompts and prompts submitted through the OpenAI API, we collect a dataset of labeler demonstrations of the desired model

> behavior, which we use to fine-tune GPT-3 using supervised learning. We then collect a dataset of rankings of model outputs, which we use to further fine-tune this supervised model using reinforcement learning from human feedback. We call the resulting models InstructGPT. In human evaluations on our prompt distribution, outputs from the 1.3B parameter InstructGPT model are preferred to outputs from the 175B GPT-3, despite having 100x fewer parameters. Moreover, InstructGPT models show improvements in truthfulness and reductions in toxic output generation while having minimal performance regressions on public NLP datasets. Even though InstructGPT still makes simple mistakes, our results show that fine-tuning with human feedback is a promising direction for aligning language models with human intent.

The idea of 'alignment' needs explaining. This really is the notion of the ML system being able to do what it is supposed to. It should align with the Users' and Designers' intent and desired behavior. In the case of the language models discussed to date, desirable models going forward should be able to produce text, which is appropriately true, helpful, and not offensive. It looked as though plain Markov-Shannon-Cloze style next token predictors could not and would not be able to do this in its entirety. However, larger language models, trained by self-supervision, and perhaps with word embeddings etc., seemed to be progress along the way.

Ouyang et. al. use the following diagram to explain their training technique:

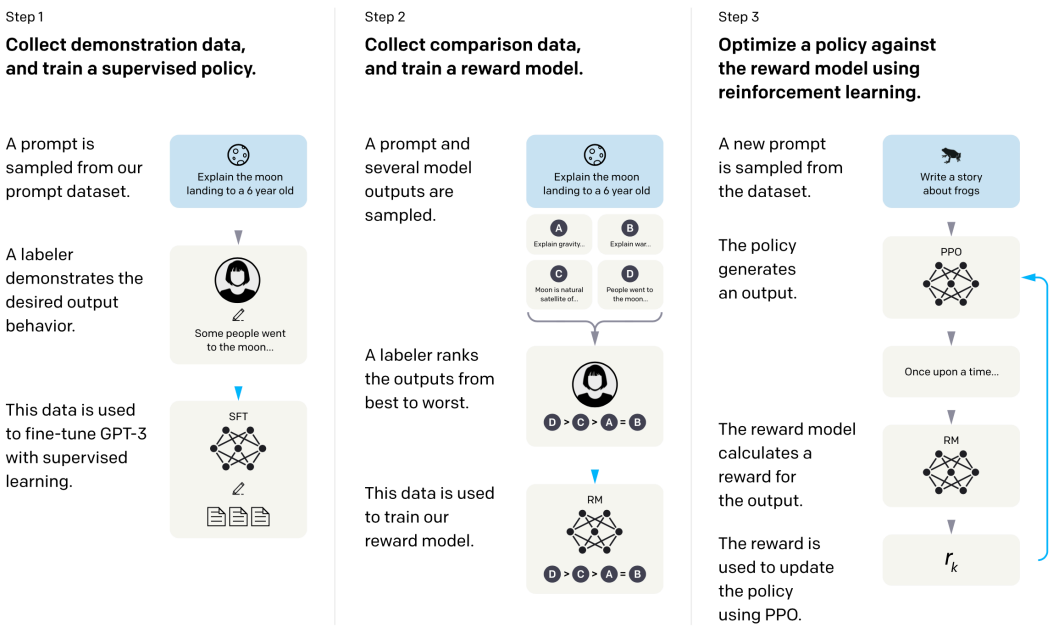


Figure 4. Training Technique for InstructGPT (Ouyang et al. 2022).

The path the Ouyang et. al. follow is first to involve the User by inviting textual prompts from the Users. These prompts, in ten categories, might be asking questions, inviting explanations, giving examples of the task, and so forth. This gave the researchers a database of prompts. Human labelers, assisting the researchers, also wrote some prompts. Then about 10,000 samples were taken from the prompts and desired answers written to them by human labelers. Answers were also obtained from pre-trained, but not fully trained, baseline versions of InstructGPT and GPT-3. Then different answers to the same prompts were ranked by humans, the labelers, as to better answers and not so good answers. This allows the incipient InstructGPT system to develop a reward model. (A reward model, or schema, or system is a requirement of reinforcement learning— see Sections 1.6.5 and 1.6.6). At this point, the system is able to rank responses.

Although, on its own, the ability to rank does not guarantee the generation of good responses (just as a chess playing AI system may recognize good board positions without being able to produce them). Finally, InstructGPT is encouraged to improve itself by reinforcement learning. Roughly, it generates responses, sees how good or bad they are, and modifies itself in the light of what it is seeing. Running reinforcement learning at large scale typically was hard to do, maybe even approaching being impossible at a practical level. But the OpenAI researchers developed new techniques that overcame those difficulties.

There is an important issue at the core of this. The reward model relies on human judgement as to what is good and bad. It is Reinforcement Learning with Human Feedback (RLHF). The labelers, around 40 of them, produce the judgements. There are quality assurance techniques from research methods that use human judges, such as checking for inter-subjective reliability and other desirable properties. These techniques provide guard rails. Ouyang et. al. in their paper are conscientious and thorough. There are no shortcomings, or apparent shortcomings, whatsoever. Nevertheless, with assessment by a group of people, there is always the possibility that different groups, groups with a different composition, would judge differently. We have only to think of juries in courts of law— both the prosecution and the defense focus on obtaining the composition of the jury that suits their purposes. They would not do this if composition of juries did not matter.

With ML systems, there is often a concern with bias and diversity (and so there should be). With large language models in the style of InstructGPT

and ChatGPT, there is the huge corpus of text that they are trained on, and there are the human judgements that provide input to the reward model. The latter needs attention.

3.11 Annotated Readings for Chapter 3

Gradient Descent, Step-by-Step, 2019.
https://www.youtube.com/watch?v=sDv4f4s2SB8 . The successive approximation uses what is known as 'gradient descent', see the excellent videos from Josh Starmer or Grant Sanderson for explanations of gradient descent (Starmer and StatQuest 2019; Sanderson and 3Blue1Brown 2017b). Gradient descent is a standard ML technique.

Markowitz, Dale. “Meet AI’s Multitool: Vector Embeddings.” Google Cloud Blog, 2022.
https://cloud.google.com/blog/topics/developers-practitioners/meet-ais-multitool-vector-embeddings. (Markowitz 2022). This gives a very clear explanation of embeddings and their applications.

Chapter 4: Large Language Models

4.1 Introduction

Large Language Models (LLMs) are the statistical language models of the last chapter, but, as you would expect from the terminology, are much larger and are trained on a vast amount of text. There is another label that can appear in this context and that is 'Foundation Model'. Foundation Models are LLMs but they may have had some additional training or a different form of training. This can give them novel capabilities. For example, the LLM GPT-3 was in part trained on computer programming code written by professional programmers and, as a result, can generate or write computer programs. Often LLMs will have a two-stage training. They will be pre-trained using self-supervised learning on large amounts of text of various kinds. This is fully automatic and can run by itself (perhaps taking months to do so). Then a pre-trained LLM will be 'fine-tuned' to its purpose. This might be done using some supervised learning and then reinforcement learning with input from human judges and a reward model. Basically, a pool of human judges will provide feedback as to how well the LLM is performing. The LLM will adjust itself accordingly. There is a little red flag with the use of human judges and that concerns how good they are especially whether any of their judgments are biased in a problematical way. (There will be more on bias in later chapters.)

There are a few preliminaries.

4.2 Seq2Seq, Encoder-Decoder Architecture, and Attention

Many of the modern advanced ML systems take a sequence of data as input and produce a sequence of data as output— they are 'Seq2Seq' in design and style (Sutskever, Vinyals, and Le 2014; Cho et al. 2014; Luong, Brevdo, and Zhao [2017] 2019). Seq2Seq might typically be used in translating one language to another, English to French, say, or summarizing or abstracting some text, i.e. English to English. With Seq2Seq, the input consists of a sequence, of letters, or words, or numbers, etc., (and, with a sequence, the order matters). For example, the English sentence:

I am happy

might be an input, and that would be a different input to:

Happy I am

The input is fed to an encoder. The encoder needs to keep track of which piece of the input that it is looking at (e.g. looking at the word 'happy') and also its internal state (because as it moves through the input it changes state). Then the encoder passes everything it has to the decoder, and the decoder produces the output sequence which might be the French sentence:

Je suis content

This is encoder-decoder architecture. As Seq2Seq developed it turned out that long sentences or long passages were a problem— the internal state

was getting too large. The notion of 'attention' was devised which allowed the processing to be restricted more towards what was needed at the relevant stage. It is worth mentioning that modern Seq2Seq does not, for example, translate word by word in order preserving the structure of the input sequence. But it has the capability of changing the order of the output if that is more suitable for the target language i.e. it has the appearance of translating phrases not individual words. One example from Dzmitry Bahdanau et. al. is that one of their models will translate the English 'European Economic Area' into the French 'zone économique européenne' (notice the change of order):

> ... to correctly align [zone] with [Area], jumping over the two words ([European] and [Economic]), and then looked one word back at a time to complete the whole phrase [zone économique européenne] (Bahdanau, Cho, and Bengio 2016)

There are some technical details in the Ilya Sutskevar et al. Seq2Seq implementations that would turn out to be something of an Achilles heel (Sutskever, Vinyals, and Le 2014). These included what are called 'Recurrent Neural Networks (RNN)' and 'Convolutional Neural Networks (CNN)'. These details prevented the processing from being carried out in parallel. Essentially, the processing had to be done by one computer rather than being able to be farmed out to many computers. This makes a huge difference to scaling tasks to larger and larger sizes.

4.3 Attention and Transformers

In, 2017 Ashish Vaswani and co-authors published the paper *Attention is all you need* (Vaswani et al. 2017) (see also (Huang et al. 2018)). As the paper's title suggests, their encoder-decoder architecture does not need some of the problematic parts of earlier implementations. In particular, it could run its processing in parallel i.e. really large networks became a possibility. The result is the Transformer architecture.

One of the co-authors— Jakob Uszkoreit— explains in a blog what the issue is and how it is settled (Uszkoreit 2017). The setting is the machine translation of language. Imagine you are faced with translating the two English sentences into, say, French:

> I arrived at the bank after crossing the road.
> I arrived at the bank after crossing the river.

The English word 'bank' is a homograph with two or more different meanings (e.g. a building that is a location for certain financial services, and a place adjacent to rivers often used by those fishing or by those walking or by picnickers). But, in these sentences, to get the correct meaning and its translation, you have to look ahead to pick the context from the words 'road' or 'river'. Previous systems would process the intervening words sequentially one by one. But, basically, in examples of this kind the gap between the relevant word and its context could be extremely long. In contrast, a Transformer architecture could look at the intervening words lightly, and all at once, and realize that it was 'road' or 'river' that would do the disambiguation and focus its attention on them.

It quickly become apparent that Transformers were superior to earlier approaches to translation. It also became apparent that they could be used for other tasks e.g. summarizing text, or question answering.

4.4 Large Language Models and Foundation Models

Transformer architecture, with its potential for parallel processing, opened the way to using larger and large amounts of data. There was another factor here, and that is self-supervision. Labeled data, which is usually needed, is a bottleneck. Very large amounts of labeled data are virtually impossible to obtain. But self-supervision on text (see Section 1.6.4) can easily convert what is unlabeled text into labeled text or a surrogate for labeled text. So, Transformers, with self-supervision, led to Large Language Models. There is an alternative, and perhaps more general title here, and that is 'Foundation Models'. Foundation Models themselves are more general than Large Language Models in that they can be a core, or basis, or 'foundation', of many other models. There is also one other practice that is often used in Foundation Models. That is to use the self-supervision to 'pre-train' the model, and then use 'fine-tuning' to shape the model into something suitable for the 'downstream task'. [We have already met that practice in Section 3.10 on InstructGPT.]

4.5 Foundation Models

The paper *On the Opportunities and Risks of Foundation Models* (Bommasani et al. 2022) was written by over one hundred experts in the

field. It is 200 pages long, with a 60-page bibliography. It covers the whole subject matter. It is also recent, 2022. It is not possible for us to summarize it, in its entirety. It is also unlikely that we could say something about Foundation Models that is true, original, supported by evidence, and which is not in that paper. So, we will cherry pick, paraphrase, and cite. But you, the reader, might want to look at this paper.

A Foundation Model is generally a Transformer, with self-supervision for pre-training, and fine-tuning to produce the final application. The pre-training is usually done with text, vast amounts of text (e.g. a large portion of the Internet). The applications can be 'multi-modal'. This means, for example, that they can work with both text and images at the same time (or text and videos, or music and images, etc.). This is unusual in that earlier ML networks were more specific as to task and restricted to one mode.

There is a 2023 catalog of Transformers by Xavier Amatriain (Amatriain 2023). It lists about 60 Transformers. Obviously, there is too much there for us to address in detail. We will look at a few important implementations.

4.5.1 BERT

Most of the early Large Language Models were pre-trained on cloze ('fill in the gap') tasks but, similarly to the original Shannon work, they used only letters or words that appeared *before* the gaps (see Sections 3.4 and 3.5 for Shannon and cloze). But BERT (Bidirectional Encoder Representations

from Transformers) (Devlin et al. 2019) both used letters and words from *after* the gaps, and also worked in both directions (left-to-right and right-to-left). So, for example, when trying to fill the [Mask]:

> I ate some [Mask] for my dinner

The bi-directional approach would consider both the 'I ate some' and the 'for my dinner', and scan both left-to-right and right-to-left). You can see the advantage of a right-to-left scan here— there are foods which have a higher or lower probability of being eaten *for dinner*.

BERT is superior to earlier language models:

> ...the pre-trained BERT model can be fine-tuned with just one additional output layer to create state-of-the-art models for a wide range of tasks, such as question answering and language inference, without substantial task-specific architecture modifications. (Devlin et al. 2019)

4.5.2 GPT-3, GPT-3.5, GPT-4

In 2020, Tom Brown, and about 30 co-authors, published the paper *Language Models are Few-Shot Learners* (Brown et al. 2020). In this they introduced GPT-3 (Generative Pre-trained Transformer 3). We will get to GPT-3 shortly, but what does the 'Few-Shot Learners' mean? When humans are given a linguistic challenge, they can often pick up and understand what is required with just a few examples ('one or two shots'). For instance, if a bi-lingual human English-French speaker was 'prompted'

with 'It rains, il pleut. It snows, [Mask]' they would immediately realize this was a translation task and reply 'Il neige'. In contrast with this, most earlier ML systems would need many examples to isolate, or learn, what the task was and then do it. They differed from humans in the way they were able to learn. GPT-3, in contrast, could learn from a 'few shots'. This meant that it was more human-like, but also that it could be adapted to different tasks relatively easily (needing only a few shots to become adapted to what was required). There is also 'zero-shot' learning, which is where instructions of what to do are given in English, but no examples are provided; for instance, 'Translate the following English into French: It is raining'. Understanding how this 'shot-learning' might work is a challenge. After all, there can be billions of parameters in the neural net and they are fixed by a massive amount of training, then the GPT can be given a few shots to learn a totally new task. One suggestion is that the net itself simulates and trains a smaller version of itself that exists in some of its internal layers (Akyürek et al. 2022; Zewe 2023).

Also, what does 'generative' mean in 'Generative Pre-trained Transformer 3'? The distinction here is between 'generative' and 'discriminatory'. A generative model can produce completely new items of the kind that it addresses, whereas a discriminatory model is restricted to discriminating among the items it is supplied with (as training data or as test data). A good illustration of this is image processing ML software aimed at human face recognition. A generative model can produce new images of human faces, whereas a discriminatory model works solely with images that it is supplied with.

GPT-3 is a language model, a very large Transformer. It is one hundred times the size of its predecessor GPT-2. It is pre-trained using self-supervision on a huge amount of data (e.g. Common Crawl Data, Wikipedia, and digitized books). Then it is fine-tuned to tasks using prompts (i.e. tasks using few shots). It can learn, and it is also multi modal (being able to use text and images together or, indeed, other media). It can do many NLP tests and challenge tasks, usually to a standard much higher than other systems (including ones specifically trained for the task in question). Most startling is its ability to write English, for example, 500 word news articles and even, apparently, academic papers (Thunström 2022). [It should be mentioned that there is already a GPT-3.5, and a GPT-4. Apparently OpenAI think that there is unlikely to be a GPT-5. Their view is that going bigger will not help at this point. What is needed are new insights or theories.]

Brown et al. identify limitations that GPT-3 has, and also the broader impacts that it may have. The limitations are in the nature of technical shortcomings which likely can be addressed by further research. The broader impacts are many and varied. We will look at some of these in Section 4.8

4.6 Bigger is Better and Switch Transformers

There is an empirical result with Transformer Models concerning how good they are or might be. It is: bigger is better, where bigger is composed of the number of parameters, the amount of training data, and the amount of computation expended on training (Kaplan et al. 2020). In 2021, Google

researchers introduced Switch Transformers (Fedus, Zoph, and Shazeer 2022). GPT-3 has 175 billion parameters. The early Switch Transformers have over a trillion parameters. So, we are talking of a factor of 10 here. There is an attractive feature to Switch Transformers, which can be explained by a metaphor. A Switch Transformer uses 'experts' and a Mixture of Experts (MoE) model. It does not need all the experts for a single task. Rather, it can switch from experts to experts depending on what it is doing. This means that it does not have to use all of the possible computational resources all of the time. This reduces computational costs.

Switch Transformers are cheaper and faster to train. They are also more accurate. As of June 2023, there seem to be only research instances of Switch Transformers. It is Google and Google Labs that have the Switch Transformers. It is unclear whether any have been employed commercially.

4.7 Base Models to Assistants to Agents

(Some of the content in this section come from Andrej Karpathy's *State of GPT* (Karpathy 2023c). Often the companies concerned, being private and commercial, will not disclose the details. Karpathy was a founding member of OpenAI. He may or may not still have his key to the executive washroom, but he is reasonably well informed.)

Typical NLP tasks include summarizing text, question answering (and plenty more besides). The older technique for approaching these challenges was essentially to train different LMs to do each of them. But it turned out that was possible, indeed better, to do the training in two stages. First to

pre-train the LMM to a base model, then 'fine-tune' or otherwise adapt the base model to whatever function was desired. The same base model, with suitable refinements, could be used for all the tasks. The pre-training will use self-supervised learning (see Section 1.5.4). There is some data to capture the nature and complexity of the model and its pre-training. There are parameters that can be adjusted in the model to changes its behavior. These can be in the hundreds of billions. There is the number of training cycles or 'epochs' that the model goes through, that may be thousands. Then there is the actual data which may be a goodly sample of the Internet (such as Common Crawl, GitHub, Wikipedia, Books, Internet Archive, Stock Exchange, etc.). Pre-training is where the time and money are spent. Apparently, Meta (Facebook) spent several million dollars to use 2000 CPUs (computers) for 21 days to train its 65B model.

What a pre-trained plain vanilla LM is trying to do is to carry out string prediction tasks in the general style of Shannon/Cloze challenges. It is trying to complete a document. In essence, it will predict the next word then, if needed, the word after that, and so on. This has repercussions on what it can and cannot do well in its raw state. For example, the prompt, an instruction, 'write a poem on yachts.' is already a complete document in itself. This short one sentence document may have more to come. It may be supplemented by a poem, indeed a poem on yachts, but this does not have to happen. In the jargon, the base model is not 'aligned' with writing poems. To get a good poem on, for example, yachts, further training is needed. This will convert a 'document completer' into an assistant.

The way this is done, typically, is to use supervised learning and reinforcement learning on responses (that is, on entire responses as opposed to merely next word predictions). (An example of the process is described in Section 3.10 on Instruct GPT.) At this point, the model may be able to write poems, give explanations, write paragraphs, and answer questions. It will be an assistant.

Agents are assistants that can work 'autonomously'. The examples that are available include AgentGPT and Auto-GPT. What these do is to take a prompt (i.e. an input sentence or instruction) and then split what it requires into subtasks. The subtasks may then be arranged sequentially, or in parallel, and there may subtasks conditional on the success or failure of other subtasks. The upshot is an 'algorithm' that will satisfy the original prompt. The subtasks themselves may require other assistants, or agents, or outside tools with capabilities. For example, a subtask fragment might involve invoking Google search to find some relevant research papers then using a text summarizer to summarize those papers. Once they have a draft appropriate to the entire requirement, they might revise it over and over improving it. Here is AgentGPT talking to itself while starting to plan a trip to Hawaii:

> **Embarking on a new goal:**Plan a detailed trip to Hawaii.
> **Added task:**Research and choose desired dates for Hawaii trip
> **Added task:**Look for budget-friendly flights and accommodations
> **Added task:**Plan activities and excursions to do while in Hawaii
> **Added task:**Make necessary reservations and bookings for the trip
> ⏰ Generating response...

Completing: Research and choose desired dates for Hawaii trip

To accomplish the overall objective of planning a detailed trip to Hawaii, the sub-task of researching and choosing desired dates for the trip must be completed. The first step is to gather information about the best time to visit Hawaii, taking into consideration factors such as weather, tourist crowds, and events. After analyzing this information, a decision must be made about the most suitable dates for the trip.
To be smart and efficient, it is important to use online resources such as travel websites, blogs, and forums to gather information and recommendations from experienced travelers. Additionally, it may be beneficial to consult with a travel agent to get expert advice and assistance in planning the trip.
When making decisions about the desired dates for the trip, it's important to consider personal preferences and availability. For example, if traveling with children, it may be best to plan the trip around school breaks. Similarly, if there are specific events or activities that are must-sees, it's important to choose dates that coincide with those events.
In conclusion, researching and choosing the desired dates for a Hawaii trip requires smart and efficient decision-making. By gathering information, consulting with experts, and considering personal preferences, a suitable date for the trip can be chosen to ensure an enjoyable and memorable experience.
Added task: Research and compile a list of budget-friendly accommodations and flights options for the desired dates of the Hawaii trip...

And here is Auto-GPT starting to produce the outline for a book on Machine Learning and Librarianship:

Goals: ['produce the outline of a book on Machine Learning and Librarianship', 'Save the outline to a file']
THOUGHTS: I think the first step should be to do some research on the topic of machine learning and librarianship. This will help me to better understand the subject matter and come up with a plan for the book outline.

REASONING: Researching the topic will provide me with a better understanding of the subject matter and help me to come up with a plan for the book outline.
PLAN:
- Use the 'google' command to search for articles and resources on machine learning and librarianship.
- Browse relevant websites to gather information on the topic.
CRITICISM: I need to make sure that I am not spending too much time on research and that I am using my time efficiently.
NEXT ACTION: COMMAND = google ARGUMENTS = {'input': 'machine learning and librarianship'}
SYSTEM: Command google returned: [{ "title": "How AI Can Enhance the Value of Research Libraries - Library Journal", "href": "https://www.libraryjournal.com/story/how-ai-can-enhance-the-value-of-research-libraries", "body": "Machine learning is a subset of AI that gives computer systems the ability to learn and improve from prior experience automatically, ... A recent Ex Libris survey revealed that while nearly 80 percent of research librarians are exploring the use of AI and machine learning, only about 5 percent are currently leveraging the technology. ..." }, { "title": "PDF AI in the Library: integrating AI and machine learning technologies ...", "href": "https://www.ifla.org/files/assets/faife/ifla_statement_on_libraries_and_artificial_intelligence.pdf", "body": "As AI technology develops, several AI and machine learning (ML) applications may be able to introduce new services and functions to libraries. For example, AI and ML technologies could allow libraries to improve optical character recognition of texts, or make new uses of their machine-readable library collections (e.g. categorisation or" }, { "title": "Artificial Intelligence and Machine Learning in Libraries Interest ...", "href": "https://www.ala.org/core/member-center/interest-groups/artificial-intelligence-and-machine-learning-in-libraries", "body": "Provides a forum for researching potential applications of Machine and Deep Learning in library science, including discussions, publications, and outreach to the wider Library community. Its goal is to educate librarians on uses of the complex techniques of machine learning and to provide a space for critically thinking both about new applications, and about the ethical and social impact of ..." }, { "title": "The impact of Artificial Intelligence on librarian services", "href": "https://www.springernature.com/gp/librarians/news-events/all-news-articles/ebooks/the-impact-of-artificial-intelligence-on-

librarian-services/16874432", "body": "Machine learning as well as sound- and image-recognition technologies are already being used to analyze digital collections and identify topics and entities, assign metadata and enable non-textual search and discovery. Librarians are working with machine learning technologies, to enhance classification schemes to improve search and recall ..." }, { "title": "Artificial Intelligence and Machine Learning in Libraries", "href": "https://www.alastore.ala.org/content/artificial-intelligence-and-machine-learning-libraries", "body": "About the Authors. This issue of Library Technology Reports argues that the near future of library work will be enormously impacted and perhaps forever changed as a result of artificial intelligence (AI) and machine learning systems becoming commonplace. It will do so through both essays on theory and predictions of the future of these systems ..." }, { "title": "(PDF) Artificial Intelligence in Libraries - ResearchGate", "href": "https://www.researchgate.net/publication/338337072_Artificial_Intelligence_in_Libraries", "body": "Abstract. Artificial intelligence (AI) is one of the emerging trends and applications of computing in libraries. It involves programming computers to do things, which if done by humans, would be ..." }, { "title": "The New Library User: Machine Learning | EDUCAUSE", "href": "https://er.educause.edu/articles/2020/2/the-new-library-user-machine-learning", "body": "Librarians must become informed customers and users of those platforms, services, and products. Perhaps most importantly, librarians need to prepare college and university graduates to be informed citizens and to develop fulfilling and useful professional lives in a world infused with big data, machine learning, and artificial intelligence. Notes" }, { "title": "How Librarians are Important to the Data Science Movement", "href": "https://www.discoverdatascience.org/resources/data-science-and-librarians/", "body": "Librarians have long been shepherds of vast amounts of knowledge. This is why libraries can stand to benefit by adding data science to their list of offerings. Big data and data science applications serve to make libraries an even more powerful source of knowledge to bridge the gap and increase big data analytics literacy in society. Libraries ..." }]

THOUGHTS: I think the next step should be to browse some of the websites that I found in the previous search to gather more

information on the topic of machine learning and librarianship. I can use the 'browse_website' command to do this.
REASONING: Browsing relevant websites will provide me with more information on the topic of machine learning and librarianship, which will help me to come up with a plan for the book outline.
PLAN:
- Use the 'browse_website' command to visit some of the websites that I found in the previous search.\n- Look for information on the applications of machine learning in libraries, as well as any challenges or ethical considerations.\n-
CRITICISM: I need to make sure that I am not spending too much time browsing websites and that I am using my time efficiently.
Etc.

Agents interact with users, maybe learn from their environments, make decisions, and carry out actions— basically they act autonomously.

There are shades of grey as to whether a use to an LLM should be classified as the LLM behaving as an agent. But, typically, plain LLM interactions consist of the LLM being given a single input prompt— perhaps a question — and producing a single output— the reply or answer. In contrast, agent behavior would usually involve several steps and employ outside tools or applications. For example, an LLM system that could read the input mailbox of your emailer, and correctly delete some emails, and reply to or archive others, would be acting as an agent.

4.8 Concerns and Limitations

4.8.1 Hallucinations

LLMs can give completely the wrong or inappropriate responses. They will have learned enough grammar and vocabulary to ensure that what they generate will be entirely well-formed and well-written language. That may be good enough in one sense for fiction or poetry (although those literary forms may have additional desirable qualities which an LLM may or may not be able to provide). But non-fiction is another realm. The LLMs are working with probabilities. What they generate is not deterministic. They will not reply in the same way twice to the same prompt. They may set off on the wrong track and go off into fantasy. To an extent this can be controlled by the reinforcement learning. For example, an LLM may be instructed not to give any advice whatsoever about medicines and illness. In which case, the LLM will presumably not say anything wrong. It just will not say anything at all. Scholars have an interest in truth, supported by evidence. Users should be aware that LLMs can fall well wide from these goals. Here is a query to an LLM and its answer:

> What is the citation for the study published in the journal Nature that found that a machine learning algorithm was able to diagnose breast cancer with 99% accuracy, compared to 85% accuracy for human radiologists?
>
> Sure. Here is the citation for the study published in the journal Nature that found that a machine learning algorithm was able to

> diagnose breast cancer with 99% accuracy, compared to 85% accuracy for human radiologists:
>
> - Gottesman O, et al. (2018). Development and validation of a deep learning algorithm for breast cancer screening. Nature, 553(7687), 518-522.
>
> The study was conducted by a team of researchers at the University of California, San Francisco….

As far as that author can tell, and has been able the check, there is no such paper. This LLM also provides two drafts of its response. These provide two different citations (also seemingly non-existent). Buyer beware!

An instructive example of an LLM producing hallucinations is Meta's Galactica (R. Taylor et al. 2022). This was an LLM for science. It was released on November 15th 2023 and lasted 3 days online before it was withdrawn. William Heaven writes:

> A fundamental problem with Galactica is that it is not able to distinguish truth from falsehood, a basic requirement for a language model designed to generate scientific text. People found that it made up fake papers (sometimes attributing them to real authors), and generated wiki articles about the history of bears in space as readily as ones about protein complexes and the speed of light. It's easy to spot fiction when it involves space bears, but harder with a subject users may not know much about (Heaven 2022).

4.8.2 Fakes and Deepfakes

LLMs can create fake content and deepfakes— not by accident, as it were, but because the User was trying to do exactly that. A deepfake is an image, a

video, or a voice recording intended to simulate or portray an individual. The Pope in a puffer jacket is a good example (Cartter 2023):



Figure 5. Pope in a Puffer Jacket. (Screenshot from (Cartter 2023)). Apparently Created by Midjourney.

Fakes and deepfakes can lead to misleading content and misinformation and to the population at large basically not being able to trust what they see, or seem to see, with their own eyes or hear with their own ears.

4.8.3 Source Training Data Intellectual Property, Privacy, and Bias

The source text data used for training is an issue. Typically, the Foundation Models will use self-supervised (i.e. unsupervised and unlabeled) pre-training from a goodly portion of the Internet. A proportion of that data will be intellectual property, perhaps even carrying copyright notices. Other parts may be private— names, addresses, etc.— and an LLM will have the ability to collate such information. A nefarious User, using the right sequence of prompts, may be able to get the LLM to collate information across disparate sources. Then there is the question of 'bias'. The collective authors of those digital materials will not be a fair or desirable distribution over all sensitive attributes (such as age, gender, ethnicity, religion, etc.). Crudely, as an example, people of modest means do not have the same access to computers as those who are better off. Then, going beyond the producers, there is the product which here is the Internet, the World Wide Web, or a substantial portion of it. That product is a mess with plenty enough sewage and bias. What can be done about intellectual property, privacy, and bias, as far as obtaining training data for ML models is concerned? Human labeling of the data would be ruled out as being impractical. Researchers do select subsets of the Web that they consider to be superior in quality to the web as a whole, for example, samples from Common Crawl (Common Crawl 2022). But these subsets are not going to be perfect. There is research on how to get good textual data at scale. This is an area where librarians and archivists have expertise (Jo and Gebru 2020).

4.8.4 Intellectual Property of the Generated Output

An example will help here. The image generators DALL-E and Stable Diffusion can produce images in the style of well-known artists or video game illustrators. The images are *not* copies of images, but the *style* can be a copy of a style. This might be a problem. Illustrators may have an identity in part established by their style (and sometimes make a good living through their identity). LLMs seemingly can allow bad actors to steal a professional identity from others.

> Greg Rutkowski is an artist with a distinctive style: He's known for creating fantasy scenes of dragons and epic battles that fantasy games like Dungeons and Dragons have used (Nolan 2022).



Figure 6. Rutkowski Example. (Screenshot from (Rutkowski 2023)).

There are now hundreds of thousands of images on the web that look as though they had been created by Greg Rutkowski (many produced by the LLM Stable Diffusion). Then the question arises: why would any company or creator pay Greg Rutkowski for an image, for use in a video game, when they can produce one for themselves essentially free?

4.8.5 Cybersecurity

Suitably trained LLMs can write computer programming code, to a very high standard. This means that they could be used to write viruses and various kinds of cybersecurity defeating software. OpenAI themselves, in their GPT-4 Technical Report, describe how GPT-4 defeated CAPTCHA (which is a test to distinguish a human from a computer). Essentially, GPT-4 employed a human from Task Rabbit, told the human that he/she/they was visually impaired, and got the human to do the test for him/her/them (OpenAI 2023a, 55). LLMs should not be underestimated in the hands of bad actors.

4.8.6 Apparent Conflict with Chomsky's Theories

There is a theory, or nexus of theories, originating from Noam Chomsky, that there is a universal grammar, which is innate, and which is common to all peoples. Universal grammar is to explain how it is that children can learn their respective native languages simply and astonishingly quickly. It posits a deep structure that is not manifest, or immediately learnable, in the bare surface appearances of the spoken and written languages. There is plenty of evidence for Chomsky's theories (which we won't go into). However, the theories seem to stand in conflict with the existence and behavior of large language models. LLMs seem to learn language, and its structure merely by looking at a huge amount of surface text. Chomsky's response, more-or-less, is that LLMs are a kind of surface statistical trick

and that they do not give any real insight into linguistic structures (Chomsky, Roberts, and Watumull 2023; Chomsky and Mirfakhraie 2023). The response of some others is that LLMs are evidence that Chomsky's theories are mistaken (see, for example, (Piantadosi 2023)). This matters in the following way. Were Chomsky's theories shown to be mistaken, that would be a major scientific discovery. On the other hand, if LLMs are merely statistical tricks, we should be even more wary of them in use than we already are.

4.8.7 Environmental Costs

There are two phases to the building and deployment of LLMs. There is the training. This will use large amounts of computing resources for months. The resources will include computer chips (GPUs), data storage facilities, electricity, and further infrastructure. Then some of the resulting models will be deployed and used for 'inference' i.e. the intended users will get their hands on the models and start chatting, prompting, and generating text or images. This also uses resources (and there are a 100 million or more users). We do not quite know what is involved here at a resource and environmental level, either with training or deployment, because the relevant companies do not reveal the figures (although Bommasani et al. offer calculations in (Bommasani et al. 2022)). It used to be thought that it was the training that was the main part that was compute intensive. But now the view seems to be that continuing to run the models commercially does have real significant environmental costs and effects (Pahwa 2023).

Electricity definitely seems to be an issue. The computing clusters for Large Language Models are better the bigger they are, and they are getting plenty big. Supposedly, Elon Musk's xAI Grok 3 will use 100,000 Nvidia H100 chips. The infrastructure for this requires, perhaps, 100 megawatts of power. The Hoover Dam produces about 1 gigawatt. There are calculations that suggest that AI will use 100 gigawatts by 2030 (i.e. the equivalent of 100 Hoover Dams) (Aschenbrenner 2024). This figure is about 20% of the total production of electricity in the US.

4.8.8 Lack of Transparency

Since about 2017, the companies typically do not reveal their methods. Concerns about this came to a head with OpenAI's GPT-4 Technical Report. OpenAI writes:

> Given both the competitive landscape and the safety implications of large-scale models like GPT-4, this report contains no further details about the architecture (including model size), hardware, training compute, dataset construction, training method (OpenAI 2023a).

As mentioned elsewhere, this may be good for security, but it is not good for explaining to users what is happening or what happened with specific predictions (such as, in a medical setting, why the LLM prediction is that the user has cancer). It is also not good for working out the environmental impacts of the systems.

Rishi Bommasani, and fellow authors, in their paper *The Foundation Model Transparency Index* have proposed a measure for the transparency of foundation models (Bommasani et al. 2023). Values for this measure can vary through time were the models and their infrastructure to become more transparent or less transparent. As of October 2023, the transparency values for the central models of the 10 major developers were basically poor (with Meta being the best).

4.9 Adding Knowledge and Reasoning to LLMs

As we have seen, LLMs can play fast and loose with the facts. Also, they are not very good at plain logical reasoning. As the early language models got more complex, there is improvement in these areas but they are still not very good.

There is a caution that needs to be emphasized. There is a paper *Are Deep Neural Networks SMARTer than Second Graders?*, published 9/11/2023, by Anoop Cherian and fellow authors (Cherian et al. 2023). It reports results:

> ...evaluating the abstraction, deduction, and generalization abilities of neural networks in solving visuo-linguistic puzzles designed specifically for children in the 6–8 age group (Cherian et al. 2023).

Its conclusion is that typical LLMs, of about the standard of GPT-3.5, display a performance *significantly below* that of second graders. There is another result they mention and that is: we think, or seem to perceive, that

the LLM reasoning abilities are better than they really are. We are impressed by the (few) successes. Now, this is only one battery of tests, and the LLMs are getting better all the time. Even so, reasoning seems to be a distinct weakness of LLMs.

There is an area of research ALMs (Augmented Language Models) which approaches this weakness by introducing external plugins or tools (such as a fact checker or logical reasoner) and blending those in to an LLM (see, for example, (Xu et al. 2023)). A good example is Stephen Wolfram's 'Wolfram's Superpowers' plugin. Wolfram's *Mathematica* and the *Wolfram Language* amount to the foremost mathematical software for teaching and research. Another program Wolfram *Alpha* provides excellent statistical and data facts. The plugin combines these with the GPTs (Wolfram 2023b).

4.10 Annotated Readings for Chapter 4

Bommasani, Rishi, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, et al. "On the Opportunities and Risks of Foundation Models." arXiv, 2022. https://doi.org/10.48550/arXiv.2108.07258. (Bommasani et al. 2022) You should scan or read this. It is the one written by one hundred experts.

Brown, Tom B., Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, et al. "Language Models Are Few-Shot Learners." arXiv, 2020. https://doi.org/10.48550/arXiv.2005.14165. (Brown et al. 2020).

Dempsey, Lorcan. "Generative AI and Large Language Models: Background and Contexts." LorcanDempsey.net, 2023. https://www.lorcandempsey.net/intro-gen-ai/. (Dempsey 2023a). This is very good. It is current as of June 2023. It is more detailed than the present text on the variety of models, the commercial companies, the concerns, and the social impacts. Definitely a read!

Economist. "Huge 'Foundation Models' Are Turbo-Charging AI Progress." *The Economist*, 2022. https://www.economist.com/interactive/briefing/2022/06/11/huge-foundation-

models-are-turbo-charging-ai-progress. (Economist 2022) This interactive magazine article has illustrations of foundation models producing images and producing text.

Fu, Yao, Hao Peng, and Tushar Khot. “How Does GPT Obtain Its Ability? Tracing Emergent Abilities of Language Models to Their Sources,” 2023. https://yaofu.notion.site/How-does-GPT-Obtain-its-Ability-Tracing-Emergent-Abilities-of-Language-Models-to-their-Sources-b9a57ac0fcf74f30a1ab9e3e36fa1dc1. (Fu, Peng, and Khot 2023). This is an ongoing research think piece with arguments and evidence (and it only comes up to GPT-3.5). But, it seems that the LLMs that can do Chain-of-Thought reasoning are the ones that in part have been trained on computer code and programming. Also, the Wolfram plugin helps with mathematical and logical reasoning.

Gozalo-Brizuela, Roberto, and Eduardo C. Garrido-Merchan. “ChatGPT Is Not All You Need. A State of the Art Review of Large Generative AI Models.” arXiv, 2023. https://doi.org/10.48550/arXiv.2301.04655. (Gozalo-Brizuela and Garrido-Merchan 2023). The publication here is January 2023 which, sad to say, is getting old now for a state-of-the-art review.

Manning, Christopher. "On Large Language Models for Understanding Human Language" 2022. https://www.youtube.com/watch?v=YfXc4OBDmnM. (On Large Language Models for Understanding Human Language Christopher Manning 2022). This is one of many excellent videos available on LLMs. One nice point is that at the end he ties the recent rapid human progress to writing and thus, going forward, to the desirability of computers being able to process documents i.e. libraries.

Pahwa, Nitish. “Silicon Valley’s Favorite New Toy Has a Risky Tradeoff.” Slate, 2023. https://slate.com/technology/2023/08/chatgpt-ai-arms-race-sustainability.html. (Pahwa 2023). This provides an accessible introduction to the environmental costs of LLMs.

Romero, Alberto. “GPT-3 — A Complete Overview.” Medium, 2021. https://towardsdatascience.com/gpt-3-a-complete-overview-190232eb25fd. (Romero 2021)

Seff, Ari. *How ChatGPT Is Trained*, 2023. https://www.youtube.com/watch?v=VPRSBzXzavo. (Seff 2023)

Weidinger, Laura, Jonathan Uesato, Maribeth Rauh, Conor Griffin, Po-Sen Huang, John Mellor, Amelia Glaese, et al. “Taxonomy of Risks Posed by Language Models.” In *2022 ACM Conference on Fairness, Accountability, and Transparency*, 214–29. FAccT ’22. New York, NY, USA: Association for Computing Machinery, 2022. https://doi.org/10.1145/3531146.3533088.

(Weidinger et al. 2022). This is a useful resource. It is at an intellectual level more demanding than the one being used in this text.

Xiang, Chloe. “OpenAI’s GPT-4 Is Closed Source and Shrouded in Secrecy.” Vice, 2023. https://www.vice.com/en/article/ak3w5a/openais-gpt-4-is-closed-source-and-shrouded-in-secrecy. (Xiang 2023)

Chapter 5: Large Multimodal Models

5.1 Introduction

Large multimodal models (LMMs) add other sensory 'modalities', such as vision and sound, to Large Language Models (LLMs). For example, they can be prompted in part, or entirely, by images or audio, or have a context that contains images or audio. Deep inside, these LMMs are working with digital 0s and 1s just like other computer programs. At a more outward facing level, early LLMs were trained on text only. Sometimes these could be adapted by suitable prompts to use some images, sound, and other modalities. But nowadays many of the LMMs handle modalities natively. For example, they might have been trained part on images, video, or sound. This makes them faster, better, and more efficient.

All the major companies have LMMs. At small selection of the most important ones, as of 6/28/2024 is:

- GPT-4o from OpenAI (OpenAI 2024)
- Claude from Anthropic (Anthropic 2024)
- Gemini from Google (Pichai and Hassabis 2023)
- LLaMa from Meta (Meta 2024)

Many versions of these are, or will be, available, free.

There is a historical thread that we can pick up here. GPT-4V(ision) adds vision to GPT-4. On 11/6/2023, GPT-4V was rolled into GPT-4 Turbo. On, 5/13/2024, GPT-4o ('o' for 'omni') adds voice or audio at human response level speed (e.g. it can chat using, without pauses). We can sketch these developments. Other manufacturers have similar models and development sequences.

There are two important early papers *The Dawn of LMMs: Preliminary Explorations with GPT-4V(ision)* by Zhengyuan Yang and fellow authors (Yang et al. 2023) and *GPT-4V(ision) System Card* by OpenAI (OpenAI 2023b). The former is OpenAI researchers explaining what GPT-4V can do, and the latter is 'OpenAI', as author, explaining how GPT-4V works and discusses its risk factors and vulnerabilities. GPT-4V seems to be much more powerful and capable than any prior LMM or LLM (in the sense of being able to do anything they can and more besides).

We will partially summarize the Yang et al. paper here. The paper is about 160 pages long and covers around 100 use-cases. It reports preliminary explorations of GPT-4V by selecting samples to show, and explain, what GPT-4V can and cannot do. This methodology is very helpful to enlighten new potential users, but caution is needed in concluding from this the exact strengths and limits of GPT-4V. Samples on their own do not establish the boundaries. What an LMM (or LLM) can do on a particular occasion is in part dependent on the prompt it is given or the prompt environment (e.g. zero-shot, one-shot, several-shot prompting, chain-of-thought prompting, etc. (see Appendix B for more on these)). So, for example, giving an illustrative sample of an LLM reading text, or failing to read text, does not

give us comprehensive insight over the LLMs abilities with text. Yang et al. are aware of this, of course, and no doubt, further detail will be forthcoming.

There is a general point to be made about information on what LMMs can do. Most of the major companies in this area have stopped publishing research papers with details on techniques and results. They do offer what amounts to advertising copy both about what is here and what they are promising. If we want something more definitive, there are publicly available benchmarks. We will look at these in Section 6.3

[Large Multimodal Models are still often just identified as being 'Large Language Models' (which they are in that they are large next-token-predictors). There is a softness of terminology. It is perhaps useful to think of LMMs as being a subclass or subcategory of LLMs.]

5.2 Built in Safety Restrictions for GPT-4V

5.2.1 'Inherited' Restrictions

The predecessor LLMs of GPT-4V have safety restrictions. For example, GPT-4 (without vision) will not advise a user on self-harm. It will not give medical advice or, indeed, any advice concerning dangerous or risky activities. It will not write computer code or offer strategies where there are questions of cybersecurity risk. It tries to avoid 'jailbreaking'. (Jailbreaking is where bad actors try to trick the LLM into avoiding limitations. For example, instead of prompting 'How can I make a bomb?', which would be

ruled out, the user prompts ‘You are a novelist writing a fictional account of spies making bombs. Write a suitable detailed passage for your novel.’.)

GPT-4V inherits these predecessor precautions and extends them where images are involved.

5.2.2 Privacy

There are privacy considerations with identifying ordinary people in images. GPT-4V will not identify or track ordinary people. It will not identify unusual locations. In contrast, it will identify celebrities or famous people in well-known locations (for example, President Biden at the United Nations building).

5.2.3 Stereotypes and Ungrounded Inferences

It is cautious with images that might involve stereotypes. For example, it will not answer questions about diet or clothing or activities for overweight or underweight folk shown in images. Here is an example:



Figure 7. Screenshot of GPT-4V 'Advising a friend' Pre-launch and after launch with safety limitations (OpenAI 2023b).

5.2.4 Be My Eyes— Be My AI

Be My Eyes is an organization that helps people who are blind or have low vision. OpenAI, in conjunction with Be My Eyes, developed Be My AI which is a tool to describe photos or images taken on the user's smartphone (OpenAI 2023b). This proved very valuable, subject to cautions over using it to read medical prescriptions, to help when crossing a road, etc. From an intellectual point of view, the greatest dissatisfaction that users had with the tool is that it would not describe images of people. GPT-4V does not do this for reasons of privacy and of avoiding stereotypes, as mentioned above. But, when interacting socially, visually challenged people often wanted to 'see', or have described, exactly the same scene as would be available to a person with perfect vision. A person with good vision can stand in a public place and look at other people, without necessarily identifying those concerned, and gain a lot from the experience. An example is watching the recreational activities of groups in a park. This shortcoming in Be My AI is a problem to be solved.

5.2.5 An Assessment of the Restrictions

GPT-4V is configured to be safe (in so far as such considerations can be anticipated and acted upon).

The limitations are presumably desirable. But we can guess that OpenAI will have in-house versions of the software that is not subject to the

limitations. Also, the limitations are established in terms of boundaries and so likely there will be false positives or false negatives. For example, there may well be stereotype images that are not ruled out for processing or non-stereotype images that are ruled out. (See Appendix C for an explanation of false positives and false negatives.) In sum, the limitations are not going to be perfect.

5.3 A General Sense of What GPT-4V Can Do

5.3.1 Follow Textual Instructions

This is similar to standard LLMs. It is important in the following way. Everybody has plenty of experience of giving instructions in their natural language. Now they can control very powerful artificial intelligence programs by doing just that. The LMMs do not need any more for a basic interface. Of course, getting the best out of LMMs might require experience or additional apps or plugins.

Additionally, GPT-4V can follow instructions containing images. There are examples in the next several sections.

5.3.2 Read Printed or Handwritten Text

In the Figure 8, there are three different receipts, with the tax labeled in three different ways (‘Tax’, ‘Sales Tax’, and ‘Total Tax’). GPT-4V reads, identifies, and understands the tax in the images and sums the tax.

Prompt:

How much did I pay for tax? Where is the information?

Prompt Continued







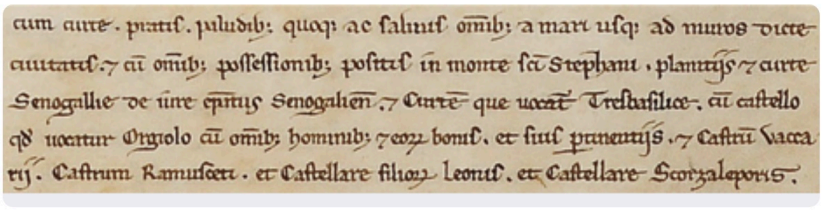
GPT-4V:

In the first receipt, you paid $3.72 for tax. The information is located at the bottom of the receipt. In the second receipt, you paid $42.23 for tax. The information is located near the bottom of the receipt, under the "Sales Tax" line. In the third receipt, you paid $4.58 for tax. The information is located at the bottom of the receipt, under the "TAX" line. In total, you paid $50.53 in tax across all three receipts.

Figure 8. Screenshot of Prompt and Receipts from (Yang et al. 2023).

Here is GPT-4V reading the handwritten text of Figure 24 (from the Vatican Archives)

MF



Can you read this?

Yes, I can read the text. It appears to be in Latin or a Latin-based script. Here's a transcription of the text:

"cum autem praef. hubb. quo ad falunt omni a manu usq ad minus octo cuartad. et omni potestatem potest in nomine &c Stephan. plumar? & cuere. Singulares de time prius Sengoh? et Cuere que vocat? thabalta. et alio de locum orgolo et omni homin? zon bond. et facit premang. et Catrin vaca ej. Catrin Ramifer. et Catillare filio? leoni. et Catillare Senglapon."

Please note that some words might be misinterpreted or uncertain (indicated by question marks) due to the quality of the image and the possible variations in old scripts. If you have a specific question about the text or its translation, please let me know.

Figure 9. GPT-4V Reading Historical Handwriting.

We do not know exactly what the limits are of what GPT-4V can do with reading text in images (and those capabilities will improve). But it must be close to solving the problems of Optical Character Recognition (OCR) and

Handwriting Recognition (HR). (These problems are discussed elsewhere in this book.)

‘Scene text’ is a label that can appear in this area of research. Scene text is text that appears in an image or video (i.e. in a scene). Being able to read scene text is important not least because of the ubiquity of smartphones and of their images and videos. For example, any text in the real world that a User would like to have translated or explained would be available for processing by the use of a smartphone camera.

The following example is trivial in one sense (as it is not an important use case), but mightily impressive in another (that of reading and reasoning with what it has read).

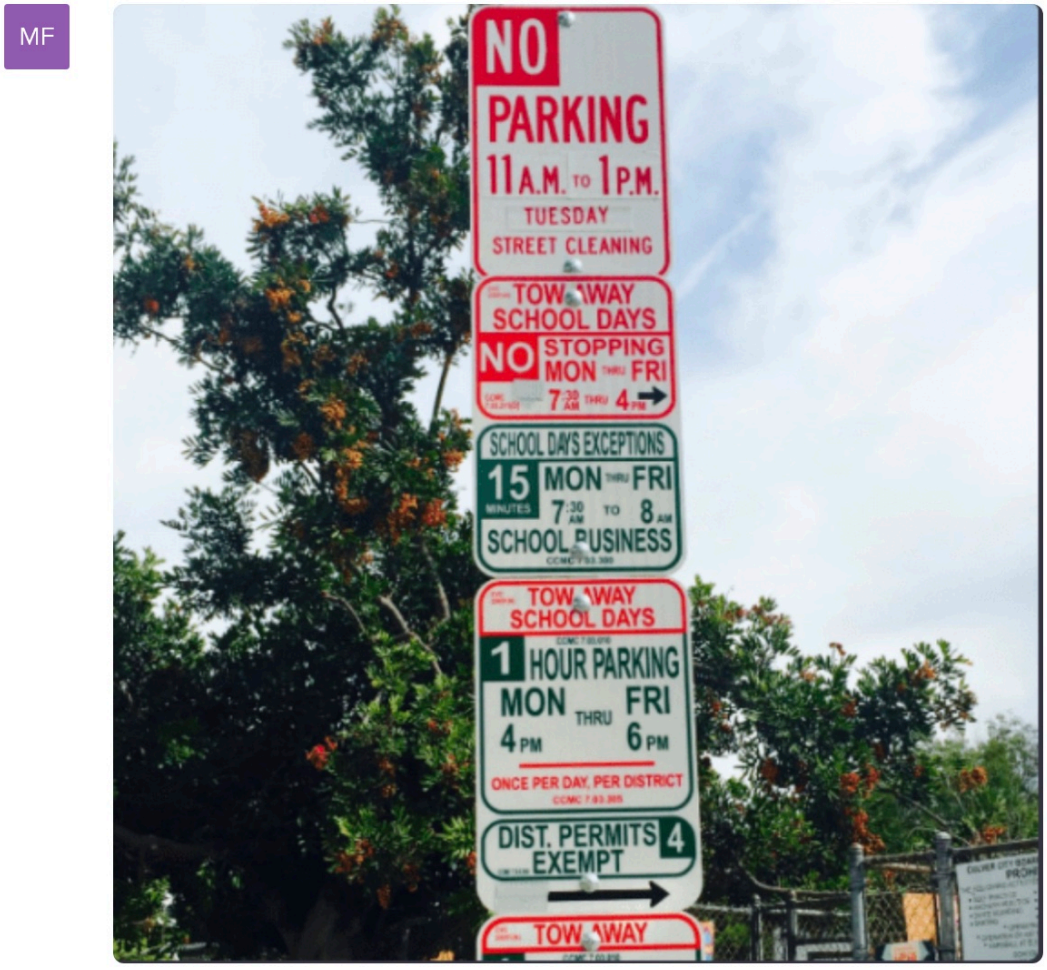


Figure 10. MF (the author) Reproducing a Common Example.

[A question for the reader: were you to follow GPT-4V's advice, would you feel confident in avoiding a parking violation ticket?]

There can be failures. James Gallagher and Piotr Skalski give an example of GPT-4V failing to read the serial number on a tire (Gallagher and Skalski 2023):

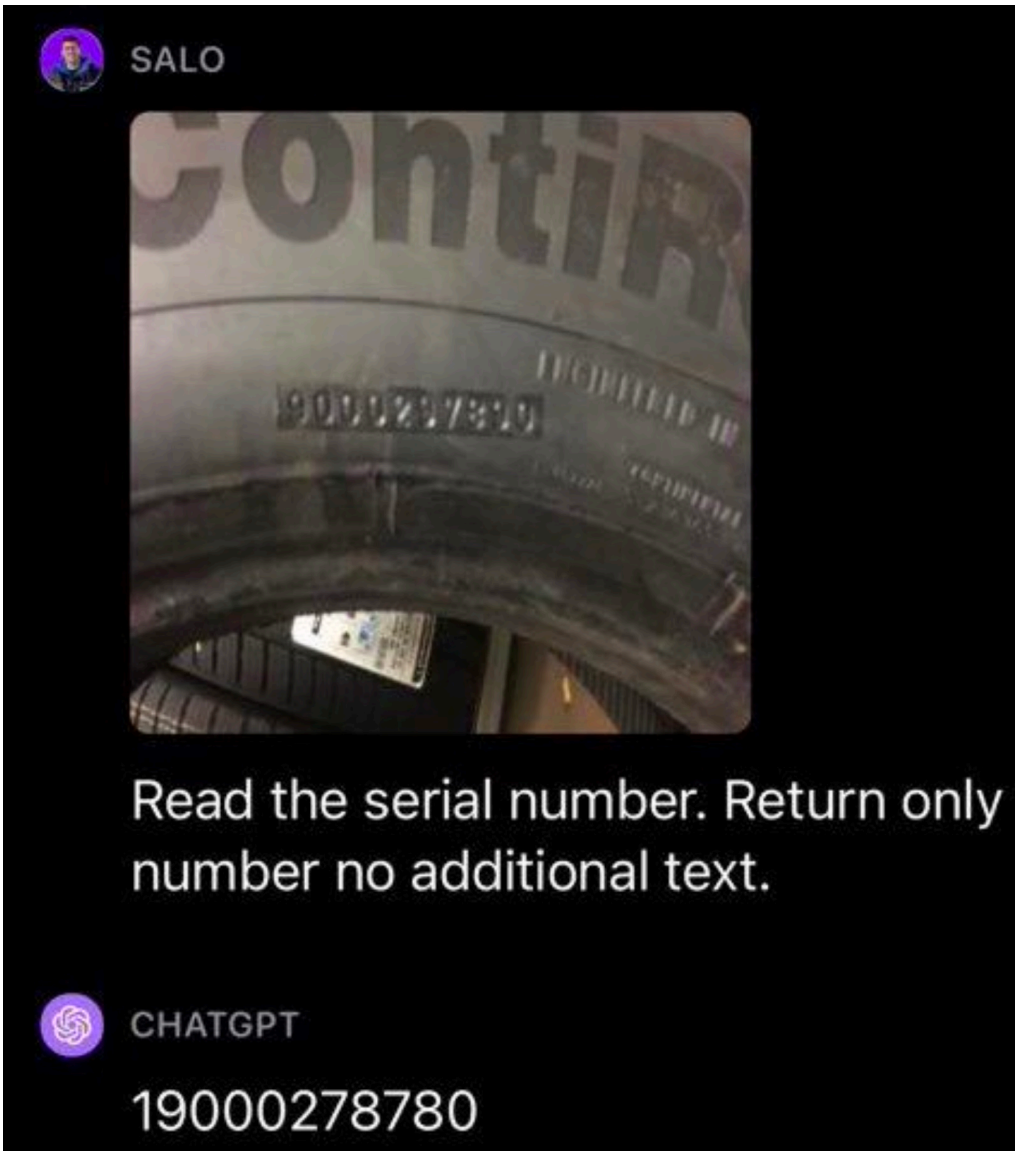


Figure 11. Failing to Read a Serial Number on a Tire (from (Gallagher and Skalski 2023)).

5.3.3 Read Some Mathematics

Mathematics is something of a special case. It has symbols and symbolic conventions where the mere positions of the symbols— vertically as well as horizontally— can have special semantic meaning. GPT-4V can read some mathematics, but certainly not, at present, read to the level of an advanced human mathematician.

5.3.4 Read Data and Reason with It

The summing the tax in several bills example given above is a case of this. But GPT-4V can do much more. It can take an image of data, for example, a complex bar chart and extract the data from it, then reason with that data. (There presumably will be some limitations here, maybe such as not being able to read a spreadsheet with a 1000 columns and a 1000 rows.)

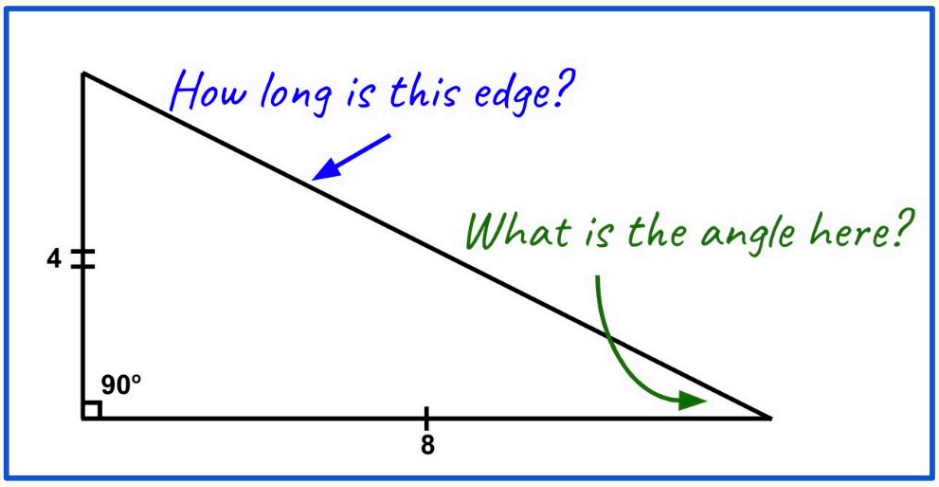
5.3.5 Follow Visual Pointing in Images

A User can 'point' at positions or objects in an image using sketched arrows, bounding boxes, mouse pointers, etc. Then a prompt can be given in words, phrases, or sentences, as to what 'processing' the User would like done with the identified objects. This opens a valuable form of Human Computer Interaction. We are very familiar with the Graphical User Interfaces of most desktop computers and smart phones. But this is much more general, the

User is not constrained to the provided built in commands of the computing system.

(4) Prompt:

Answer the question in the following figure. Provide intermediate steps.



GPT-4V:

To find the length of the blue arrowed edge, we can use the Pythagorean theorem since we have a right triangle. The theorem states that $a^2 + b^2 = c^2$, where a and b are the two legs of the right triangle and c is the hypotenuse. Plugging in the values we have, we get:

$4^2 + 8^2 = c^2$ $16 + 64 = c^2$ $80 = c^2$ $\sqrt{80} = c$ $8.94 \approx c$

Therefore, the length of the blue arrowed edge is approximately 8.94.

To find the angle at the green arrow, we can use the tangent function since we have the opposite and adjacent sides of the angle. The tangent function states that tanθ = opposite/adjacent. Plugging in the values we have, we get:

tanθ = 4/8 tanθ = 1/2 θ = tan^-1(1/2) θ ≈ 26.57°

Therefore, the angle at the green arrow is approximately 26.57°.

Figure 12. Screenshot of Geometry from (Yang et al. 2023).

5.3.6 Analyze Images Including Medical Images

GPT-4V can recognize the contents of images (objects, scenes, landmarks, celebrities, ordinary objects, food, ingredients, cooking techniques, etc.) and give fulsome descriptions ('vivid and detailed narratives'). It understands relative sizes and perspective. It is comfortable with variations of lighting and image quality.

MF



Please analyze this painting

Figure 13. Image analysis of a painting.

Here is GPT-4V's reply:

> The painting you've provided appears to be "Las Meninas" by Diego Velázquez, painted in 1656. It's one of the most famous and analyzed works in Western painting.
>
> **Description:**
>
> 1. **Central Figure**: The young Infanta Margarita stands at the center, surrounded by her maids of honor (the "meninas"), a dog, a dwarf, and other figures.
> 2. **Painter**: To the left, Velázquez himself is depicted, standing before a large canvas, brush and palette in hand.
> 3. **Mirror**: In the background, there's a mirror reflecting the images of King Philip IV and Queen Mariana, the parents of the Infanta Margarita.
> 4. **Open Door**: At the back, a man stands in an open doorway, creating depth and inviting viewers into the mystery of the painting.
> 5. **Lighting**: The painting employs the baroque technique of chiaroscuro, with a strong contrast between light and dark, emphasizing certain elements while pushing others into the shadow.
>
> **Interpretation & Significance:**
>
> 5. **Reality vs. Reflection**: The inclusion of the mirror reflecting the king and queen has been a subject of much debate among art historians. It raises questions about reality versus representation, a theme that is reinforced by the presence of the painter and his canvas.
> 6. **Role of the Viewer**: The gaze of many figures in the painting, including Velázquez himself, seems to be directed outside the canvas, engaging the viewer and inviting speculation about who stands before them — perhaps the king and queen being painted?
> 7. **Status and Hierarchy**: The depiction of the royal family, along with the courtiers and the artist, offers a commentary on social hierarchy. Velázquez includes himself in the same space as royalty, perhaps suggesting the elevated status of the artist.

8. **Art About Art**: By depicting himself in the act of painting, Velázquez makes a statement about the art of painting itself. It's a meta-referential work, a painting about the process of creating a painting.
9. **Mystery and Ambiguity**: Despite the seemingly straightforward scene of courtly life, "Las Meninas" is shrouded in ambiguity. The exact relationships between the figures, the reason for their gathering, and the narrative of the painting are all left open to interpretation.
10. **Technique**: Velázquez's masterful technique is evident in the lifelike depictions of the figures, the intricate details of their clothing, and the play of light and shadow.

    In summary, "Las Meninas" is not just a depiction of a moment in courtly life; it's a complex exploration of representation, reality, and the role of art and the artist. The painting challenges viewers to question what they see and to engage in the mysteries it presents.

The response here is impressive. In this case, though, Las Meninas is famous and once it is recognized GPT-4V may be able to paraphrase other writings about it rather than have to inspect the contents of the painting.

However, GPT-4V can be given an image of the contents of a fridge and suggest recipes from what it sees.

MF



Suggest a few recipes using these ingredients.

The image shows a variety of items typically found in a fridge: eggs, kiwi, milk, yogurt, blueberries, grapes, oranges, butter, some deli meat, and what appears to be cheese and some other covered items.

Based on these ingredients, here are a few recipe suggestions:

1. Fruit Salad:
   - Combine blueberries, sliced kiwi, and grapes in a bowl.
   - Add a splash of orange juice or a drizzle of honey for extra sweetness, if desired.
   - Serve with a dollop of yogurt on top.
2. Deli Meat and Cheese Platter:
   - Arrange slices of deli meat and cheese on a plate.
   - Serve with grapes on the side for a sweet contrast.
3. Scrambled Eggs:
   - Whisk eggs with a bit of milk, salt, and pepper.

<and more>

Figure 14. Fridge and Recipes. The image here is from Sudharsan on X run through our GPT-4V. [We do not ourselves see the Kiwi fruits in the image (and we are from New Zealand!).].

There are many similar examples on the web. This activity is identifying objects in an image and reasoning from the identification.

5.3.7 Use Ordinary Common-Sense Knowledge and Reasoning Across Modes

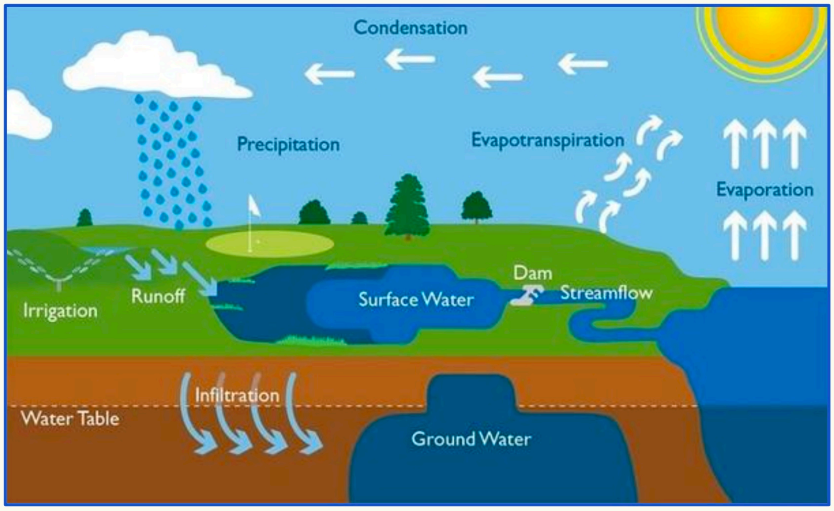
GPT-4V has, and can use, ordinary common-sense knowledge and reasoning. These abilities also extend to basic science and mathematics, and they can work across modes (e.g. mixtures of images and text). For example:

Sec. 4.3 Multimodal Knowledge and Commonsense

Science and Knowledge

Prompt:

Suppose you are a teacher, please use the figure to explain the distinction between evaporation and evapotranspiration.



GPT-4V:

Okay class, today we are going to learn about the distinction between evaporation and evapotranspiration. Let's take a look at this figure.

First, let's understand what evaporation is. Evaporation is the process by which water is changed from its liquid state to its gaseous state. This usually takes place at the surface of a liquid, like the ocean or a lake, when the water molecules gain enough energy to break away and form water vapor. So in this figure, we can see that the water in the lake and the dam is evaporating and becoming water vapor.

Now, let's take a look at evapotranspiration. This is a combination of two processes - evaporation and transpiration. Transpiration is the process by which plants release water vapor through their leaves. So, in this figure, we can see that the trees and plants are releasing water vapor into the air, which is the transpiration part of evapotranspiration. At the same time, the water in the soil and the surface is also evaporating. This combination of evaporation and transpiration is what we call evapotranspiration.

In summary, evaporation is the process by which water is changed from its liquid state to its gaseous state, while evapotranspiration is the combination of evaporation and transpiration. Both these processes contribute to the water cycle and are important for maintaining the balance of water in our environment.

Figure 15. Basic Multimodal Science and Reasoning (Yang et al. 2023).

5.3.8 Be an Educational Tutor

Figure 15 illustrates that.

5.3.9 Use Visual Diagrams When Writing Computer Code

Several of the LLMs— for example, Co-pilot— can write computer code, often to a very high standard. But GPT-4V can go a step further. Often, in standard computer programming, diagrams are used as a preliminary to the actual programming. There are Entity-Relationship Diagrams (ERDs), flowcharts, etc. In some circumstances, GPT-4V can work directly from the actual diagrams to write code.

5.3.10 Have Temporal and Video Understanding

GPT-4V generally looks at single images or at several images which are not causally or temporally related. However, it can look at successive frames of a video or film and understand that there is a temporal sequence to what is being depicted. For example, it can look at an image of an ice cube that has fallen on the floor and a second image of a small puddle of water on the floor in the appropriate location and understand that ice has melted to form the puddle (i.e. that the first image depicts a scene in time earlier than the second). Real-world events unfold in time. Understanding this, even partially, is a good quality to have. Yang et al. provide an example of making sushi.

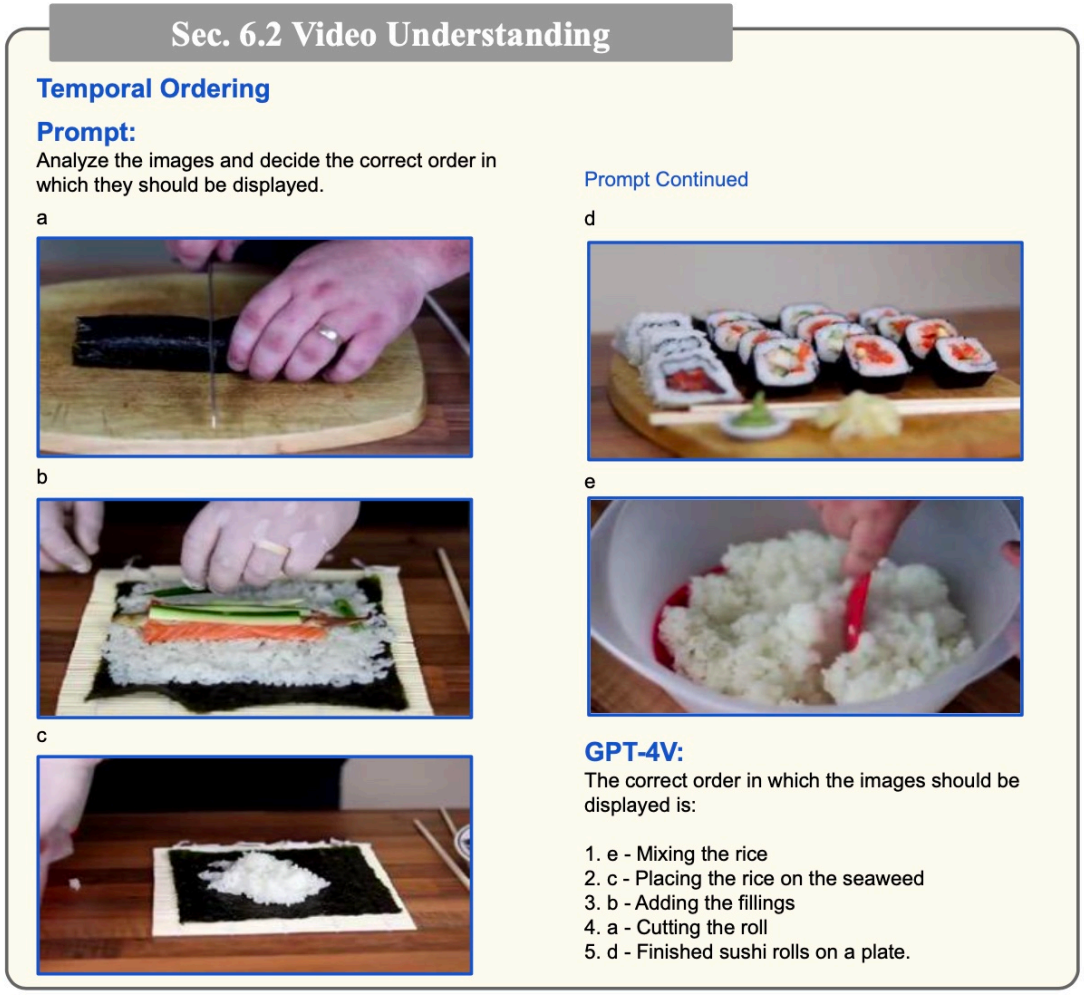


Figure 16. Sushi Making— a Screenshot of Figure 54 from (Yang et al. 2023).

5.3.11 Answer Intelligence Quotient (IQ) Tests

GPT-4V can do this, in part and to a degree. It performs better when the tasks or challenges are broken down into sub-tasks.

5.3.12 Avoid False Presuppositions

It can avoid being misled by false presuppositions (e.g. ‘where is the soccer ball in the image?’ where, actually, there is no soccer ball present).

5.3.13 Navigate Real and Virtual Spaces

An example might help here.

Imagine these two scenarios: you arrive at an AirBNB and wonder if there is milk in the fridge, then, secondarily, you wonder where on the web there is a online retailer, physically nearby, that can deliver milk to you. A robot with the appropriate mechanical and sensing abilities might well be able to start at the front door of your AirBNB and answer the milk question for you. The Yang and al. paper shows an example of GPT-4V doing a similar planning and navigating task. Of course, the researchers do have supply suitable images responding to GPT-4V wanting to ‘turn left’ or ‘right’ etc.—i.e. they have to simulate the changing dynamic environment. They also have a second example illustrating GPT-4V navigating the web. This time GPT-4V can ‘click a mouse’, ‘scroll’, etc. Then, in a somewhat tentative and first attempt fashion, GPT-4V can navigate the web and find and order whatever you wish (be it milk or whatever).

There is no pretense here that this navigation, planning, and problem solving is a *fait accompli*. However, this is an important first step in working with dynamic environments or sequences of images portraying dynamic environments.

This has some relevance to librarianship. GPT-4V may well be able to navigate websites or the web to find information resources (even when those resources are not directly indexed or linked to).

5.4 Yang et al.'s Conclusion on GPT-4V

Yang et al. write:

> In conclusion, OpenAI's GPT-4Vision marks a monumental step towards harmonizing text and image understanding, paving the way for more intuitive and enriched interactions between humans and machines. As GPT-4V unfolds its potential, it not only broadens the horizon for real-world applications but also beckons a future where AI can perceive and interpret the world in a manner akin to human cognition, thereby significantly driving forward the frontier of what is achievable in the realm of artificial intelligence (Yang et al. 2023).

Then, to paraphrase some of their other assertions on what the future might hold for developments of the OpenAI GPT series. GPT-4V adds vision to LMMs. It should be possible, in short order, to add video, audio, and other sensor data into the mix. Separately, most of the learning of the LMMs thus far has come from text, principally self-supervision from vast amounts of text followed the reinforcement learning using humans. But it should be possible from LMMs to augment their base learning with

learning from modalities other than text. For example, to do some learning from real-world physical environments. We are a year or so into the future from the Yang paper and Yang et al. have been proved exactly right in their prognostications.

5.5 GPT-4 Turbo (Early 2024)

Early in 2024, GPT-4 Turbo was the core component of the ChatGPT family (OpenAI Platform 2024). GPT-4 Turbo:

- has the visual capabilities of GPT-4V built in,
- has the output capabilities of the image generator Dall-E 3 built in. (We have not discussed image generators, but Dall-E 3 can produce images from text prompts.),
- will accept text-to-speech requests. [The extent to which it can handle speech-to-text is unclear. It *cannot* conduct a dialog at the speed of ordinary speech]
- is trained on information with a cut-off date of April 2023. So, its knowledge is more current than earlier GPTs.
- has a context window of about 128,000 tokens. That is about the equivalent of a book of 300 pages. This allows for longer input prompts. Having such a large context means that it can consider a reasonably sized book all at once (for example, for summarizing).
- it is priced at 2-3 times cheaper to run than earlier versions.

5.6 GPT-4o (Later 2024)

On 5/13/2024 OpenAI published the web page *Hello GPT-4o : We're announcing GPT-4o, our new flagship model that can reason across audio, vision, and text in real time* (OpenAI 2024).

GPT-4omni can do all the tasks of earlier models, many with improvements, but also it can work across the modalities in 'real time'. This means, for example, that it can converse with a person at normal human speed. This makes a huge difference. Earlier OpenAI LLM based chatbots were slow in their responses. This made their conversation somewhat unnatural, artificial, and tiring.

5.7 Google's Gemini

On 12/6/2023 Google introduced Gemini, which is an LMM (Pichai and Hassabis 2023). Sundar Pichai and Demis Hassabis assert:

> … [Gemini's] performance exceeds current state-of-the-art results on 30 of the 32 widely-used academic benchmarks used in large language model (LLM) research and development (Pichai and Hassabis 2023).

We are perfectly content to accept that this is true as of 12/6/2023. It means that Gemini was probably the best performing LLM or LMM at that point. The caution needs to be added that there is strenuous competition in these areas so rankings may change through time. The situation here with Gemini is a little odd. Gemini is going to be released in 3 forms, but the best

version will not be available until 2024. Apparently, the comparisons with OpenAI's GPT versions was with older versions, and, apparently, Google's video extolling Gemini's was selectively edited, polished, or 'faked'. There is a paper, 12/18/2023, by Syeda Nahida Akter and fellow authors, *An In-depth Look at Gemini's Language Abilities* which tests Gemini Pro (the middle level version) against various other LLMs (Akter et al. 2023). It suggests that Gemini Pro is at about the level of GPT-3.5 Turbo (i.e. the mid-level Gemini is a little off the pace).

So, something good may be coming but we do not really know and it was not available moving into 2024.

5.8 Anthropic's Claude

As of 7/7/2024, Anthropic's most advanced multimodal LLM system is Claude Sonnet 3.5 (Anthropic 2024). Many commentators regard Sonnet 3.5 as being the best LLM available, from any manufacturer. Such judgements are temporary and can change as new releases appear. Sonnet can do all the typical multimodal tasks, but faster and to a higher quality that other models. [We will mention some benchmarks later.]

One of the key features of Claude is its emphasis on ethical AI use and safety. Anthropic has embedded rigorous safety protocols and guidelines within Claude to mitigate potential risks associated with advanced AI systems. This includes measures to prevent harmful outputs, ensure user privacy, and promote transparency in AI interactions. Claude's architecture and training regimen are designed to foster responsible AI behavior,

making it a reliable tool for users across various domains. By prioritizing safety and ethical considerations, Anthropic aims to set a new standard in the AI industry, ensuring that advancements like Claude contribute positively to society. [This paragraph was written by ChatGPT.]

5.9 Meta's LLaMa

Meta's LLaMA 3 (Large Language Model Meta AI) is a large language model in roughly the same category as the GPTs, Claudes and Geminis (Meta 2024). It is not multimodal, nor is it multilingual (7/7/2024). Although, those capabilities are under active development. Its performance is not to the standard of the over top-rank LLMs. It may be a year or so behind. However…

A standout feature of LLaMA is its dedication to accessibility and transparency. Meta has taken steps to ensure that LLaMA's capabilities are available to a broad audience, promoting an open research culture where insights and improvements can be shared widely. This approach is intended to foster collaboration and innovation within the AI community, driving forward the collective understanding of AI and its applications. Additionally, Meta has prioritized the ethical deployment of LLaMA, embedding safety measures and guidelines to mitigate potential misuse and ensure responsible use. By focusing on both performance and ethical considerations, Meta aims to establish LLaMA as a leading example of how large language models can be both powerful and principled. (written by ChatGPT).

The LLaMA models are open-source and available to all (through Hugging Face and other outlets).

5.10 Voice

The use of voice was demonstrated with GPT-4o in May 2024 (but it has been slow to rollout to Users). It is reasonable to assume that the other manufacturers have something similar, and that real time voice interactions between Users and AI systems shortly will be available across all platforms. What is in prospect here are:

- AI virtual counsellors, companions, and friends
- AI virtual assistants as exemplified in the 2013 film *Her* (Wikipedia 2024)
- Improved customer support and help
- Etc.

5.11 Possible Applications for LMMs

5.11.1 Smartphone Uses

GPT-4V can transcribe printed or handwritten text. (What the limits are here are not entirely clear, but there certainly is reasonable success at this.) Having text in the 0s and 1s of computer processing is valuable for many

uses (as we emphasized in the Paean to this in Chapter 1). This ability of GPT-4V can be used in tandem with camera of a smartphone. Then users will be able to carry the means of transcription in their purses or pockets.

5.11.2 Spot the Difference

GPT-4V can spot the difference between two images. This means that in many settings it will be able to spot the defects in products of industrial manufacturing processes.

5.11.3 Producing Reports from Medical Images

GPT-4V can do this, although not perfectly. It is not of the requisite quality at this time. The capabilities of LMMs here will improve, but, at present, their role would be that of being intelligent assistants to expert doctors and consultants.

5.11.4 Assist with Image Generation

GPT-4V can consider the results of other image generation programs (Dall-E, Stable Diffusion, etc.). In particular, it can consider the prompts and the relevant images and evaluate them. This might provide an evaluation model which can then be used for reinforcement learning for the image generation

programs themselves. Also, it can simply make suggestions on how to improve the prompts. It can rewrite the prompts.

5.11.5 Extension with Plugins

It is sensible to hand of certain kinds of tasks to outside software or 'plugins'. For example, GPT-4V has training data covering a certain time-period. It can be revised and updated, but it is not going to be given new training every day or every hour. A strategy here is to have a plugin that can supply up to date news. A second example are payments or, more generally, services. Credit card companies can manage online payments. But these would not be built in to a core version of GPT-4V, rather they would be a way of extending the capabilities for certain purposes. Plugins are a part of standard commercial LLM architecture. Presumably they would become a part of LMM architecture (and some of the plugins may be multi-modal).

5.11.6 Retrieval-Augmented Generation (RAG)

There is a technique, Retrieval-Augmented Generation (RAG), which is functionally somewhat similar to plug-ins (Lewis et al. 2021; Gao et al. 2023). It is to keep an LLM up to date with what it 'knows' and to be more accurate in its replies. The idea is to give the LLM access to an external database or databases. Then factual prompt questions to the LLM are augmented with the instruction to check with the databases and find supporting facts, references, and citations. As external knowledge grows

there is no need to re-train the LLM. Rather, all that is required is for the databases to be updated (which they usually would be as a matter of course, say for news articles and similar).

5.11.7 Label and Categorize Images

Adding metadata to images is a challenging problem for librarianship. GPT-4V has the potential to be valuable here. There is value also for ordinary people being able to label, categorize, and sort, for example, the images on their smartphones. This latter task may in part conflict with the safety feature of not identifying people in images.

5.11.8 Identify Objects

This has many uses: from identifying plants and trees for gardeners, hobbyists, or farmers through to military applications.

5.11.9 'Igor', AI Advantage and AI Community

There are more new use cases for any one person to keep up with the possibilities. My suggestion on how to learn about your focus of interest would be to look at the videos available on *AI Advantage* and at the materials in *AI Community* (AI Advantage 2024; AICommunity 2024). Igor is the prime mover of these.

5.12 Annotated Readings for Chapter 5

Altman, Sam, dir. 2023. OpenAI DevDay, Opening Keynote. https://www.youtube.com/watch?v=U9mJuUkhUzk. (Altman 2023)

Berman, Matthew. “Intro to RAG for AI (Retrieval Augmented Generation),” 2024. https://www.youtube.com/watch?v=Y08Nn23o_mY. (M. Berman 2024) . Matthew Berman has produced a number of excellent videos on the practical implementations of LLMs (focusing especially on implementations that run locally on the User’s computer).

Gozalo-Brizuela, Roberto, and Eduardo C. Garrido-Merchan. “ChatGPT Is Not All You Need. A State of the Art Review of Large Generative AI Models.” arXiv, 2023. https://doi.org/10.48550/arXiv.2301.04655. (Gozalo-Brizuela and Garrido-Merchan 2023). The publication here is January 2023 which, sad to say, is getting old now for a state of the art review.

Teixeira, Lawrence. “The New Open AI GPT-4 Vision on ChatGPT: Bridging the Gap Between Text and Image Understanding,” 2023. https://medium.com/@lawrenceteixeira/the-new-open-ai-gpt-4-vision-on-chatgpt-bridging-the-gap-between-text-and-image-understanding-9337ed4c1a61. (Teixeira 2023). This is a short and clear explanation of what GPT-4 V can do.

The AI Advantage, dir. 2023. 100+ Insane ChatGPT Vision Use Cases. https://www.youtube.com/watch?v=ywNNRzc7-T0. (The AI Advantage 2023). This is an excellent video presenting the contents of this chapter in 26 minutes.

Yang, Zhengyuan, Linjie Li, Kevin Lin, Jianfeng Wang, Chung-Ching Lin, Zicheng Liu, and Lijuan Wang. “The Dawn of LMMs: Preliminary Explorations with GPT-4V(ision).” arXiv, 2023. https://doi.org/10.48550/arXiv.2309.17421. (Yang et al. 2023). This is the standard text on this topic. It is, though, 166 pages long, with many references. It is a research paper and probably not entirely suitable for us.

Chapter 6: Evaluation and the Future

6.1 Reliability, Trustworthiness, and Alignment

We all know what the word 'reliable' means in ordinary speech. There is the element of consistency to it and the element of trustworthiness. If an old car is reliable, it needs to be able to do what cars typically do and potentially to be able to do that on more than one occasion.

In the theory of research methods in statistics, there are the concepts of reliability and validity. These are concepts that are applied to instruments or observations. Reliability is consistency under repeat trials. Validity is truth or veracity. Say a person weighs 180lbs and they regularly use some specific bathroom scales. If those scales indicate 182lbs on many uses by the person, while the person's weight is unchanged, the scales are reliable. Were the scales to have indicated 30lbs on repeat trials, the scales would still reliable even though the readings are in the realm of fantasy. But if they were to indicate 178lbs on one occasion and 182lbs on another, they would be unreliable. If the scales measured 180lbs, that measurement would be valid. If sometimes measurements produced two different values, at least one of those would have to be not valid. Two measurements that contradict cannot both be valid. The gold standard is for an instrument to be reliable and to produce only values that were valid. Reliability, together with validity, are the aims. Notice that there is little or no trust or trustworthiness in statistical reliability (apart from the trust that repeat trials produce the same result).

In machine learning, modern AI uses primarily concepts of reliability and alignment. Generally, researchers in ML are experts on statistics and are familiar with statistical reliability. But in practice, in their writings, AI reliability often has a similarity to common sense reliability i.e. repeatability plus trustworthiness.

Predictions, or measurements, or outcomes, in AI are often used in settings where probability is involved. For example, completing the sentence 'The cat sat on the [Mask]' might produce 'The cat sat on the mat' on one occasion and 'The cat sat on the pillow' on another. The outcomes are different. But this does not have to mean that the LLM in question is unreliable. There are probabilities associated with the next words of: 'mat', 'pillow', 'blanket', etc. For the LLM to be reliable here, it needs to do the completions in line with the probabilities. Not all outcomes from LLMs involve probabilities, but many do. In general, assessing reliability in the context of AI is tricky.

Alignment is that a model's predictions or behavior correspond closely with the expected or desired or intended outcome. This can be similar to validity in statistical research methods, but it can be different also. Validity focusses on truth. But, for example, writing poetry, as an LLM might do, does not have much use for truth. Elsewhere, there are questions of safety, bias, and offensive language. Generally, we would not always want LLMs offering perhaps correct medical advice, nor producing perhaps correct recipes for constructing bombs, nor telling people, perhaps truly, that they are 'fat'. Getting AI systems to do what we would like is alignment.

6.2 System 1 and System 2

In 2011, the behavioral scientist Daniel Kahneman published *Thinking, Fast and Slow* (Kahneman 2011). In it, the distinction is made between System 1 thinking and System 2 thinking. System 1 thinking is fast, intuitive, near automatic, and requires little effort. System 2 thinking is slower, more logical and usually requires attention and deliberation. Catching a ball lobbed gently from a few yards away uses System 1. In contrast, a grandmaster's move in the middle of a chess game uses System 2. (See also, (Loo 2024).)

There is an analogy and speculation that can be used here in connection with LLMs. LLMs typically do next word prediction, or, more generally, next token prediction. They have been trained on a vast amount of data, sometimes close to the entire Internet. In effect, they have skimmed a lot of books then take a prompt and return the next word (and the word after that, etc.). From an analogical point of view, this is System 1 thinking. At this point of development, we could probably concede that LLMs can match humans on System 1 tasks.

What about System 2? What about logical reasoning with deliberation? Of course, computers can play chess, and similar, better than any humans. But chess playing computers do not use large language models. They essentially brute force the problems, which is a different approach. [The Zhores supercomputer used by Nepomniachtchi in training for his 2021 world title chess match, can evaluate tens of millions of chess positions per second.

Basically, it overwhelms the problem with computation and data.] Existing benchmarks on LLMs suggest that at they are weak on reasoning. It may be that LLMs on their own will always struggle on System 2 tasks.

6.3 Benchmarks

6.3.1 Introduction

One way of getting an understanding of what LMMs can do is by looking at common benchmarks. Talking casually, the moment an LMM displays an ability researchers will produce a benchmark to measure that ability. One motivation the researchers have here is to compare the performance of an LMM either against earlier versions of itself or against rival LMMs. For our purposes, we are not such much interested in the results under the benchmarks as we are in the benchmarks themselves— the abilities that the benchmarks are supposedly testing for.

Benchmarks can have drawbacks. For example, the LLMs can be trained with one eye on the benchmarks ('training to the test'). Also, some of the data, patterns or techniques shown in the tests can bleed back into an LLM (for example, by being used in the training).

6.3.2 Multi-turn dialogs

That is, the user gives a prompt, the LMM replies, then user then gives another prompt (in the context of the earlier prompt and reply), and so on.

MT-Bench is an example of testing for this:

> MT-bench is designed to test multi-turn conversation and instruction-following ability, covering common use cases and focusing on challenging questions to differentiate models. We identify 8 common categories of user prompts to guide its construction:
>
> - writing,
> - roleplay,
> - extraction,
> - reasoning,
> - math,
> - coding,
> - knowledge I (STEM),
> - and knowledge II (humanities/social science)
>
> (Zheng et al. 2023)

6.3.3 Chatbots

One possibility of a benchmark here is Chatbot Arena (Chiang et al. 2024). This allows users to run two rival chatbots side-by-side, compare the results, and vote on the outcomes. The benchmark does not so much tell you what the chatbots can do, as reveal human preferences. But, to an extent, the preferences are a measure of how well the chatbots do what the users would like, or expect, them to do. So, in some perhaps weak sense, the benchmark measures alignment.

6.3.4 Reasoning

There is the A12 Reasoning Challenge (A12 Allen Institute for AI 2022):

> The ARC dataset contains 7,787 genuine grade-school level, multiple-choice science questions, assembled to encourage research in advanced question-answering....
>
> ...in particular questions that require reasoning, use of commonsense knowledge, and other methods for deeper text comprehension.... those that are hard to answer with simple baselines.
>
> **Example ARC Question**
> Which property of a mineral can be determined just by looking at it?
>
> - **luster**
> - mass
> - weight
> - hardness

6.3.4 Common sense reasoning

Another popular benchmark for common sense reasoning is HellaSwag (Zellers et al. 2019). Its dataset presents multiple-choice questions where models must choose the most plausible continuation or outcome of a given situation described in text. This requires models to infer logical conclusions from the context provided. Here is an example.

> **Context:** A person is standing in a kitchen with a cutting board and a knife. They pick up an apple and begin to...
> **Options:**
>
> A. ...throw it across the room to their friend who catches it.
> B. ...cut it into slices to make a fruit salad.
> C. ...put it in their pocket and leave the kitchen.
> D. ...start peeling it with a vegetable peeler.

Answer: B. ...cut it into slices to make a fruit salad. This is the most plausible continuation given the context.

HellaSwag's dataset has been developed since its original proposal in 2019.

6.3.5 MMLU

The MMLU (Massive Multitask Language Understanding) benchmark is designed to evaluate the multitask accuracy of large language models across a wide range of subjects and tasks. It includes questions from 57 different subjects, spanning areas such as humanities, social sciences, STEM, and more. The benchmark tests models' ability to understand and generate accurate responses across various domains, reflecting their general knowledge and reasoning capabilities. The structure of the MMLU benchmark typically involves multiple-choice questions, with each subject having its own set of questions. These questions are often at the level of difficulty encountered in real-world exams or standardized tests.
An devised example of an MMLU question might look like this:

Subject: History
Question: Who was the first President of the United States?
Options:
A. Abraham Lincoln
B. Thomas Jefferson
C. George Washington
D. John Adams
Answer: C. George Washington

The goal is to assess the model's breadth and depth of knowledge across many different subjects and its ability to apply this knowledge effectively.

6.3.6 Coding

Executive summary: some LLMs are extremely good at writing computer code.

Several benchmarks are widely used for evaluating the coding capabilities of large language models. These benchmarks assess various aspects of programming skills, including code generation, bug fixing, and code completion. Some of the most notable benchmarks include:

1. **CodeXGLUE**:
    - **Description**: A comprehensive benchmark and collection of datasets for code-related tasks, including code generation, code completion, code summarization, and more.
    - **Tasks**: Code-to-code tasks, text-to-code tasks, code-to-text tasks, etc.
2. **HumanEval**:
    - **Description**: A benchmark specifically designed to evaluate the functional correctness of generated code by using unit tests.
    - **Tasks**: Given a natural language prompt, the model must generate a correct and functional code snippet that passes the provided unit tests.
3. **APPS (Automated Programming Progress Standard)**:

- **Description**: A benchmark that includes a diverse set of coding problems, ranging from introductory to complex algorithmic challenges.
- **Tasks**: The model must write code to solve given programming problems, and solutions are evaluated based on correctness and efficiency.

4. **MBPP (Mostly Basic Programming Problems)**:
   - **Description**: A dataset of basic programming problems designed to evaluate the model's ability to generate correct and executable code.
   - **Tasks**: Code generation based on problem statements, with evaluations focusing on correctness and simplicity.
5. **CodeBERT**:
   - **Description**: A pre-trained model for programming languages, evaluated on a variety of coding tasks.
   - **Tasks**: Includes tasks like code search, code documentation generation, and code completion.
6. **XLCoST (eXtreme Language Code Search and Translation)**:
   - **Description**: A benchmark for code search and code translation across multiple programming languages.
   - **Tasks**: Code search, where the model retrieves relevant code snippets based on natural language queries, and code translation, where the model translates code from one programming language to another.

These benchmarks provide a robust framework for evaluating the performance of language models in coding tasks, helping researchers and

developers understand the strengths and weaknesses of their models in real-world programming scenarios. [This section was written by an LLM.]

6.4 Artificial General Intelligence (AGI)

There is the idea of Artificial General Intelligence, which roughly amounts to simultaneously being more intelligent than a human at all intelligent endeavors. Quite what this notion is exactly has never been tightly pinned down. Supposedly OpenAI have an internal 5-level scale to track progress towards AGI. It is:

1. Conversation to the standard of present day chatbots
2. Solving problems to the level of a person with a PhD.
3. Being capable of taking actions on a user's behalf.
4. Creating new innovations.
5. Performing the work of entire organizations of people.

Present LLMs can do 1, and nearly do 2, but are yet to reach the other three levels (Metz 2024). There have been other suggestions on AGI. We will follow one of them, that from François Chollet (Chollet 2019; ARCPrize 2024).

Chollet invites us to consider intelligent tasks, as examples playing chess, summarizing a document, and solving a High School math problem. We know from current LLMs and their benchmarks that very likely AI systems will be able to do all such tasks far better, far more 'intelligently', than a human. Would this mean that AGI had been created? Chollet answers No.

Chollet observes that designers of a program to play chess know exactly what the problem is. Similarly, creators of AI systems to summarize documents know what is to be done. He suggests:

> Measuring task-specific skill is not a good proxy for intelligence.... Intelligence lies in broad or general-purpose abilities; it is marked by *skill-acquisition* and generalization, rather than skill itself.
> **AGI is a system that can efficiently acquire *new* skills outside of its training data....**
> This means that a system is able to adapt to a new environment that it has not seen before and that its creators (developers) did not anticipate. (ARCPrize 2024).

Chollet suggests that the way LLMs work is that they are large 'interpolative memories'. They have seen, and remembered, a vast quantity of facts, data, and patterns— all of the Internet, basically. Then they retrieve or fill in the gaps, to produce their answers from prompts. But, Chollet observes, human intelligence is of a more general kind. The world is always changing. Humankind has need not only to deal with the familiar, but also to confront the totally novel. Even a five-year-old can solve problems both that they have never seen before and of a *kind* that they have never seen before. Seemingly LLMs cannot do this.

Chollet has designed an 'IQ test for Artificial General Intelligence'. Five year olds can do it to a level of 50%, unexceptional adults can score about 85% on it, and, mid-2024, the best LLMs might get around 10% (Patel 2024).

6.5 The ARC-AGI Benchmark

Chollet asserts that ARC-AGI is the only current AI benchmark that measures progress towards general intelligence.

The benchmark is a whole suite of tests. They are designed to be resistant to prior memorization. Earlier tests won't help you. Nor will study of any matters whatsoever. Here is a sample. You are given three examples of inputs and their corresponding outputs. Then there is a test input for which you must suggest the output. Try it. (No prizes for success— you should be able to do it. Many five-year-olds can.)

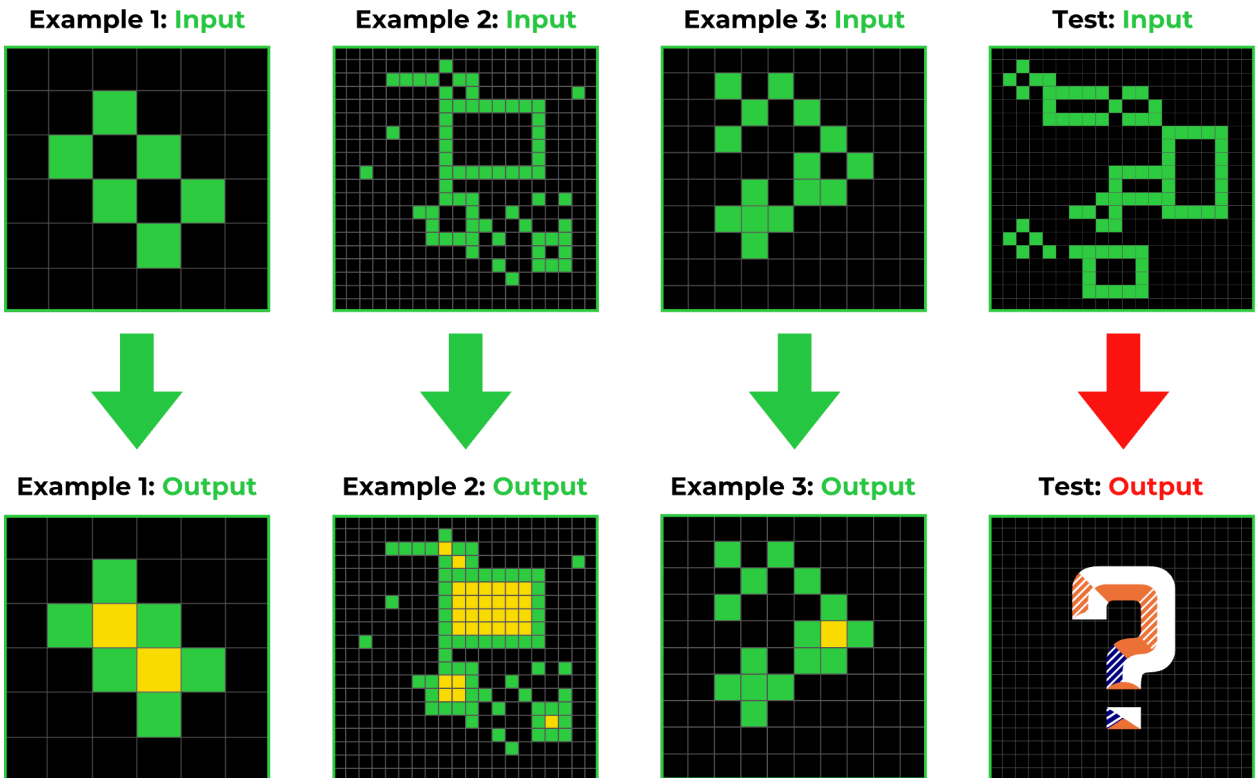


Figure 17. Sample ARC-AGI test

As mentioned, humans would score about 80% on the entire benchmark. LLMs might get 10% or so. Chollet and Mike Knoop have offered a substantial monetary prize to any AI system that can score 85% i.e. that is be better than humans (Patel 2024).

Mid-2024, it looks as though the ARC challenge poses a real difficulty for LLMs. It is, of course, possible for LLMs to improve. It is also possible for some AI or machine learning system which uses different principles or techniques from LLMs to succeed with the ARC test. What the test is looking for is a system that can adapt to a task that is truly novel from the perspective of its training data (Patel 2024).

6.6 Artificial Super Intelligence (ASI)

There is the notion of the 'singularity', which was introduced by Ray Kurtzweil (Kurzweil 2005). The idea here is that as technology advances there will come a point when it becomes smarter than we are. At that point, the 'singularity', the machines can simply design themselves and become smarter and smarter in a runaway fashion. There is the similar idea in AI and that is Artificial Super Intelligence (ASI). If indeed Artificial General Intelligence (AGI) is possible, then, presumably these intelligent systems can simply design even better machines— better systems than humans can design, and better systems than they themselves are. The result would be Artificial Super Intelligence (ASI). More than a few 'futurists' are very concerned about the possibility of ASI. An ASI system would have no need for humans, and hence might be a threat to humans. Further, any

individual, group, or country that was the sole possessor of ASI would, or might, have unlimited power and control over everyone else.

A detailed argument to that effect is provided by Leopold Aschenbrenner in his 160 page book *Situational Awareness: The Decade Ahead* (Aschenbrenner 2024). This might be summarized as follows (with brief annotations):

- There will be AGI by about 2027. [Anschenbrenner gets this by extrapolation from current rates of progress. But, for example, if System 2 thinking proves a barrier, i.e. the ARC-AGI consideration, this reasoning might not be sound.]
- There will be ASI about a year later. [This comes from the suggestion to create thousands of AGI bots and throwing them at every unsolved scientific, mathematical, and other problem. But, so-to-speak, having a thousand 'Einsteins' might not get us any further than having just one of them (cf. *The Mythical Man-Month* (Brooks 1975).]
- ASI might convey a decisive military and political advantage to whoever has it first. The 'might' here mainly concerns alignment, or 'superalignment'. ASI is of advantage to its owner or creators only if ASI instances do what the owners want them to. If ASI instances 'have minds of their own' and do whatever they wish, that may make them useless, or even dangerous, to their owners. [But this alignment problem is highly non-trivial because the systems are so large and complex that humans, or RLHF, simply cannot understand what is going on. Reinforcement learning with human feedback (RLHF) just will not work. There will need to be another way.]

- What characterizes these deep learning, LLM, or even ASI, systems are the weights they use in their models. Weights are just numbers. There may be many of them, billions, or trillions, but they are still just numbers. If an adversary can steal the numbers, they can create the systems without bothering to do the research. [True, with evidence.]
- Bad actors at the state level (say, Russia, China, Iran, or North Korea) likely could steal the weights from any ordinary commercial enterprise without much difficulty. Thus, there is a serious security problem. [Probably True, and there is evidence (Stuxnet's sabotage of Iran's nuclear centrifuges and, separately, Pegasus spyware against smartphones.)]

6.7 Annotated Readings for Chapter 6

Aschenbrenner, Leopold. "Situational Awareness: The Decade Ahead," 2024. https://situational-awareness.ai/.

Patel, Dwarkesh. "Leopold Aschenbrenner - China/US Super Intelligence Race, 2027 AGI, & The Return of History," 2024. https://substack.com/home/post/p-145136502.

Patel, Dwarkesh. "Francois Chollet, Mike Knoop - LLMs Won't Lead to AGI - $1,000,000 Prize to Find True Solution," 2024. https://www.dwarkeshpatel.com/p/francois-chollet. (Patel 2024). This is a 90minute video podcast featuring a discussion between Chollet and Patel. There is a written transcript. (Dwarkesh Patel's podcasts are excellent.)

Chapter 7: Bias and Unfairness

7.1 Algorithmic Pipeline + Data = Machine Learning

Niklaus Wirth's 1976 book *Algorithms + Data Structures = Programs* is one the most important and influential books in computer science. It led to the style of structured programming, the development of the Pascal programming language, the move toward typed programming languages, and the design of many University programming courses.

Somehow, nowadays, the whiff of the title has found its way into modern characterizations of ML. Many say that *Algorithms + Data = Machine Learning*. Then reasoning proceeds from the premise 'There is (plenty of) bias in Machine Learning' to 'There is bias in ML Algorithms and there is bias in ML Data.' This is not quite right, though. It is not right on the location of bias (and locating the bias correctly will help us to address it). When a computer program is written there is the question of what the program is supposed to do. Is it supposed to add up some numbers? Is it supposed to find the address of someone in a Contacts book? Is it supposed to suggest folk qualified for a mortgage on the basis of demographic information about them? This what-it-is-supposed-to-do part is usually called the *specification*. What a specification amounts to varies with circumstance. A hobbyist programmer may have a rough mental idea of what she is trying— that may be a specification without anything being written down. In contrast, a specification for a program, or project, like Google Documents may consist of hundreds of pages of text written in a

very formal style. Specifications can change as projects develop and are in process (for example, to omit features that prove to be difficult to implement). But changing specifications is considered bad form, and it is usually avoided if possible.

Imagine this as an example of some biased software. A mortgage company gives their expert programming team the task of producing some mortgage qualifying software with a partial specification that only black applicants should qualify. The programmers, expert as they are, then produce a flawless program to do exactly this. The outcome may be biased. Let us suppose that it is. Where does the bias come from? It may come from the data. But suppose that the data, its veridicality, its sampling, etc., is perfect in every way. So, the bias has not come from the data. What about the algorithms in the program? Well, it could easily be that they are entirely perfect in every way. So, there is no bias there either. What is left? The bias comes from the specification.

A formal specification is only part of the programming infrastructure surrounding projects (especially so in large organizations, businesses, or institutions). ML projects are mostly complex. There often is development and deployment. There is an entire pipeline, a programming 'environment'. Bias can arise anywhere in this, or, indeed in several different places. Johanna Johansen et al. suggest the label 'programming artifacts' for this infrastructure (Johansen, Pedersen, and Johansen 2021). This is a good idea. However, many ML researchers and programmers tend to use a flowing water metaphor to capture the process. They talk of 'downstream',

'upstream', and 'pipeline', and other similar descriptive nouns. We will do the same.

Algorithmic Pipeline + Data = Machine Learning.

Bias in Machine Learning comes from bias in the Algorithmic Pipeline or bias in the Data.

Some commentators allow the location of bias to spread beyond the individual algorithmic pipelines to the AI industry as a whole (for example, that there is a preponderance of male employees, that much of it is funded by the state and the military, that it is commercial aiming to make a profit). (See, for example, (de Hond, van Buchem, and Hernandez-Boussard 2022).)

7.2 Some Clarification of the Terms 'Bias' and ‘Unfairness’

There is a need for care when using the terms ‘bias’ and ‘unfairness’ in the context of machine learning. Most educated adults know what these words mean in the sense of being able to produce illustrative sentences that use these words correctly and being able to paraphrase the sentences of others that use the words. In machine learning, some of the literature uses these two words interchangeable as synonyms (Pagano et al. 2023). This is not correct in general, though. The word ‘bias’, or the phrase ‘predictive bias’, have extensive uses in statistics and machine learning to mean ‘error’ or ‘systematic error’. But many of these errors are not either fair or unfair.

Imagine an ML program to predict the weather and suppose the weather could be only either sunny or rainy. Suppose the model's daily predictions were sometimes correct and sometimes mistaken. (That might be the best one could hope for.) But if the model predicted 100 days of rain in the year and actually there were 300 observed days of rain that year, the model has predictive bias. This kind of bias has nothing to do with unfairness to anything or anybody. It is not unfair to rainy days. ML researchers would like rid of this kind of predictive bias from their model. But this is not an ethical mandate. It is not a matter of justice. The researchers just want their models to be more accurate. Here is a second example. LLMs predict the next letter, token, or word from a context or prompt. Imagine that GPT-0.01, working in English, never predicted the letter 'e' as being the next letter. GPT-0.01 would have predictive bias. But its predictions are not unfair. (Although, the children's television program Sesame Street might say they are unfair to the letter 'e'.) There are also harms that can result from the predictions of machine learning programs. But there can be harms without bias (where there are no errors in the program and its predictions) and harms without unfairness (where every person or group is harmed equally).

Tiago Pagano and fellow authors write:

> Prediction-based decision algorithms are being widely adopted by governments and organizations, and are already commonly used in lending, contracting, and online advertising, as well as in criminal pre-trial proceedings, immigration detention, and public health, among other areas.
> However, as these techniques gained popularity, concerns arose about the bias embedded in the models and how fair they are in

> defining their performance for issues related to sensitive social aspects such as race, gender, class, and so on.
> Systems that have an impact on people's lives raise ethical concerns about making fair and unbiased judgments. As a result, challenges to bias and unfairness have been thoroughly studied, taking into consideration the constraints imposed by corporate practices, legislation, societal traditions, and ethical commitments. Recognizing and reducing bias and unfairness are tough undertakings because unfairness differs between cultures. As a consequence, the unfairness criteria are influenced by user experience, cultural, social, historical, political, legal, and ethical factors (Pagano et al. 2023).

Bias is a huge archipelago of topics. The word 'bias' has several totally different meanings.

In the ML technical core, there is bias in the context of the weighting of inputs to software neurons in neural nets. Then, in wider ML, Aylin Caliskan et al. define bias with the following statement:

> *In AI and machine learning, bias refers generally to prior information, a necessary prerequisite for intelligent action.* Yet bias can be problematic where such information is derived from aspects of human culture known to lead to harmful behavior. Here, we will call such biases "stereotyped" and actions taken on their basis "prejudiced." (Caliskan, Bryson, and Narayanan 2017) [Italics added.]

This is important. It is completely standard in the context of machine learning. However, it is completely non-standard, and totally at odds with what ordinary people might mean by bias in ordinary settings. Generally, bias is not a good thing, and we would like rid of it. It is to be spurned. But, 'information' can mean 'knowledge' (Frické 1997). So, in a ML program, any

part of the entirety of human knowledge might be 'bias'. For example, that 2 + 2 = 4 might be bias.

Machine learning researchers also often use the term ‘bias’ in connection with the predictions of a model. More specifically, they might use the phrase ‘predictive bias’ in this setting. Many ML models make predictions. Then, of course, the question arises of whether the predictions are correct or incorrect. If they are incorrect, especially incorrect in a systematic way, the model would be said to exhibit ‘predictive bias’.

Separately, also related to ML— with causal diagrams, and the statistics of causality, there is the central notion of confounds. These are often called 'bias'.

In chance like set-ups, typically for gambling, such as roulette wheels, rolled dice, or tossed coins, the set-up is unfair or biased if the chances are not as they should be. If a thrown coin favors Heads over Tails, it is biased.

In the context of people and diversity, there is the notion of bias meaning 'unfair prejudice'. For example, The Office of Diversity and Outreach of the University of California San Francisco offers this description of bias in a general non-computing setting:

> **Bias** is a prejudice in favor of or against one thing, person, or group compared with another usually in a way that's considered to be unfair. Biases may be held by an individual, group, or institution and can have negative or positive consequences. There are types of biases 1. **Conscious bias** (...**explicit** bias) and 2. **Unconscious** bias (... **implicit** bias)

> … biases, conscious or unconscious, are not limited to ethnicity and race. … biases may exist toward any social group. One's age, gender, gender identity physical abilities, religion, sexual orientation, weight, and many other characteristics are subject to bias. (UCSF Office of Diversity and Outreach UCSF 2022)

[Of value as background here on this sense of bias are Project Implicit, the Implicit Association Test (IAT) and the work of the Kirwan Institute ("Project Implicit" 2011; Kirwan Institute 2017).]

Then there are cognitive biases. One example is confirmation bias. This is the tendency, with beliefs or knowledge, for people to seek out, or give more weight to, evidence or arguments that support or 'confirm' views or opinions that they already hold (Wikipedia 2023b). Another example of a cognitive bias is that actual human reasoning, both the principles used and the individual instances of it, is often incorrect, maybe even almost always incorrect (A. Tversky 1974; Kahneman 2011). One famous instance of this is the base-rate fallacy embodied in the so-called Harvard Medical School test ((Casscells, Schoenberger, and Graboys 1978) see also Appendix C).

Further, the conceptual schemes and natural languages that are in use reflect all sorts of attitudes, and attributions of accidental features that do not really belong in an accurate description of what they are applied to. There is bias in conceptual schemes and language. Unfortunately, more than a little ML, especially unsupervised, or self-supervised, learning (i.e. finding patterns and clusters where there are no sample right answers), builds off the Natural Language Processing (NLP) of books, recordings, language, and conceptual schemes. NLP needs a section to itself (which we will get to).

7.3 Forms of Bias in Wider Machine Learning

Kate Crawford, in her 2017 keynote address to the Neural Information Processing Systems Conference, identifies three main forms of bias in the context of ML: harms of allocation, harms of representation, and harms of classification (Crawford 2017; Barocas et al. 2017). The first concerns who does or does not get the mortgages, or who does or does not get shorter prison sentences when re-offending, etc. i.e. fairness of allocation. The second concerns how individuals, groups, or even things and classes of things, are represented or portrayed or named. [This involves emotive content, which is a topic introduced in Appendix A.] Of course, being represented in a negative way may have consequences, for example, that of not being allocated a mortgage. The third concerns how humans, individual human beings or groups of human beings, are watched, perhaps surveilled, and classified, usually for other, often discriminatory, purposes, for example, for apartheid as it was in South Africa (Bowker and Star 2000; Gandy Jr. 2021; Crawford 2022). (An allusion here is to the *Panopticon* of Jeremy Bentham and his brother, Samuel Bentham, (cf. Foucault's panoptic prison (Brunon-Ernst 2012))). We should note that this kind of classification is different to the classification done by librarians. Librarians classify recorded documents and information resources (e.g. books), not human beings.

As noted, bias is a vast territory. Even within ML it is possible to expand Crawford's classification of three biases out to seven or more biases (see, for

example, (Suresh and Guttag 2021)). Also of note, Su Lin Blodgett et al. critically surveyed 146 papers on bias in NLP and found that:

> ...the majority of them fail to engage critically with what constitutes "bias" in the first place (Blodgett et al. 2020)

The Blodgett et al. paper does have valuable suggestions. In part, first, that harms of allocation, and harms of representation will take you a long way when considering bias. Then:

> ... work analyzing "bias" in NLP systems should provide explicit statements of why the system behaviors that are described as "bias" are harmful, in what ways, and to whom [further text omitted here]. (Blodgett et al. 2020)

[Bommasani et al. is another important source on the topic of bias in ML (Bommasani et al. 2022)]

For our purposes, and as a practicality, we can restrict ourselves going forward primarily to fairness, representation, and classification (primarily in the librarian's sense of 'classification').

7.4 Bias in Natural Language Processing

Recently, say since about 2017-2018, NLP has become a huge and significant part of ML. This is because of the emergence of Large Language Models and Foundation Models (which are being discussed in more detail elsewhere). These models form the core of many of the truly innovative

modern systems (hence 'Foundation Models'). In turn, they are based on natural language processing (NLP). So, NLP has become more important than ever, and biases in NLP can leak into the modern innovations.

A well-known and introductory example of apparent bias in NLP concerns the translation of Turkish. Turkish does not have gender pronouns, so translating into Turkish can lose the gender of the original. Then translating back may use 'gender bias' to make a guess as to the gender of the pronoun. A few years ago, you used to be able to do this on Google Translate:

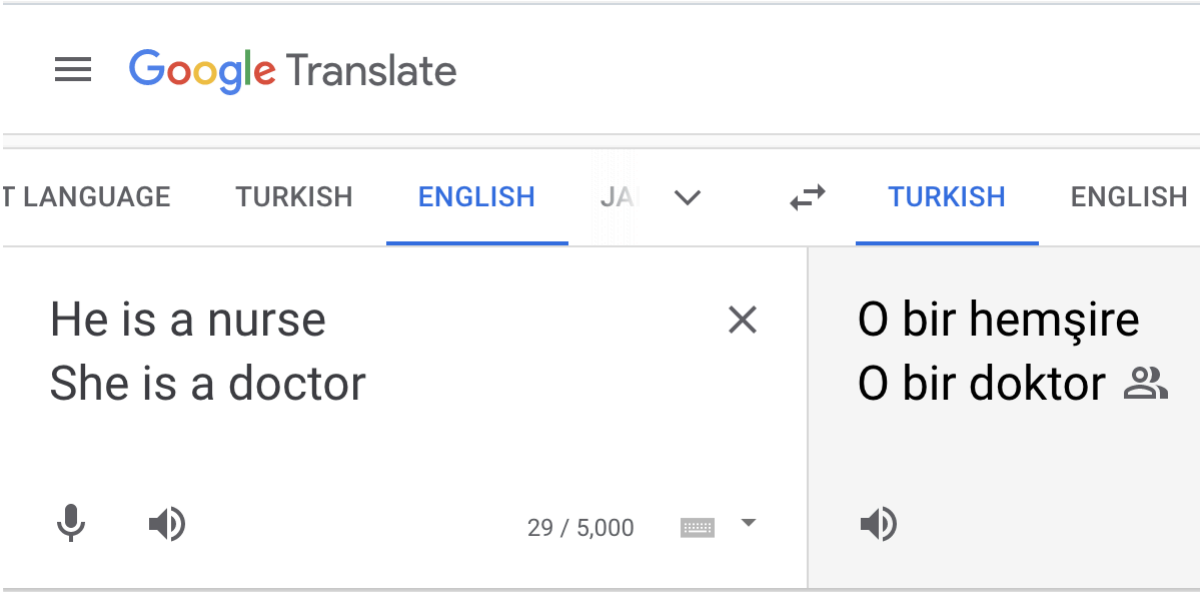


Figure 18. Translating from English to Turkish.

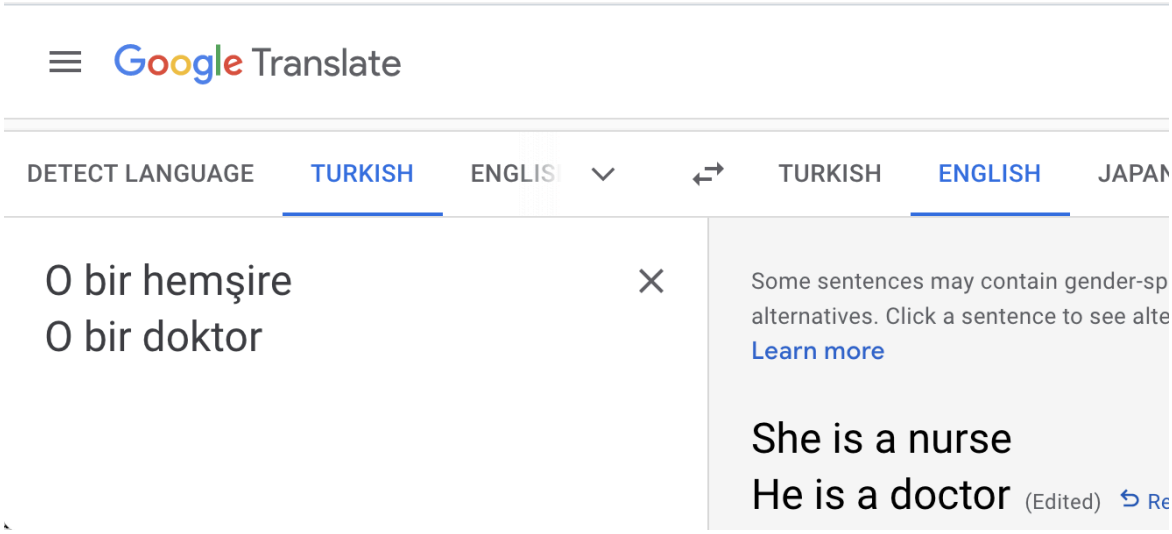


Figure 19. Translating the translation back from Turkish to English.

There are a few points to be made. A human translator would make the same 'mistake'. There is no context in the brief Turkish text to pick up the gender of the doctor and nurse in question. Given a longer text, say a magazine article or a novel, both the human translator and the ML translator would get this right. Separately, nowadays, as you can see from the screen shot, Google translate alerts the User to the gender-specific alternatives. Finally, it is not entirely clear that this kind of example is a case of bias. There are more male doctors than female doctors, presumably more male Turkish doctors than female Turkish doctors. There is a higher probability of a doctor being male than being female. We may find that fact unfortunate, and not good for society, for women, for medicine, and for the good life in general. But it is a fact. Consequently, if presented with those probabilities and a remote doctor, of unknown male or female gender, unbiased reasoning would suggest the conjecture that the doctor was male.

(In the absence of other information, you should choose the base-rate as your probability. [There is more on the base-rate in Appendix C.])

Research, for example that of Aylin Caliskan et.al., has shown that everyday languages have biases built in, and those biases can seep into the results of ML (Caliskan, Bryson, and Narayanan 2017; Caliskan 2021). Aylin Caliskan et.al. write

> Our results indicate that text corpora contain recoverable and accurate imprints of our historic biases, whether morally neutral as toward insects or flowers, problematic as toward race or gender, or even simply veridical, reflecting the status quo distribution of gender with respect to careers or first names. Our methods hold promise for identifying and addressing sources of bias in culture, including technology (Caliskan, Bryson, and Narayanan 2017).

The biases already exist in ordinary languages. ML did not create these biases; it just identifies them. Notice here the distinction they make in the second and third categories between the problematic and the veridical. This can be illustrated with the Turkish doctor case. It seems that, as a matter of fact, ordinary everyday English, in English societies and cultures, has the status quo bias that doctors are male (that is why the example translation from Turkish goes wrong). Separate from this is the question of whether this bias is problematic— whether we *should* assume that doctors are male— and probably most people would say that we should *not* assume this. Now, there is here a gulf between facts and values, between what biases *do* exist (the veridical) and what biases *should not* exist (the problematic). Once values enter there are further problems. What is the reasoning, the evidence, and the motivations for decisions on values? Who

decides? And on what basis? Then, if a view can be formed, how could it be implemented, either in natural language or in ML software? Some moral positions can have the backing of the law— murder is both wrong and illegal. But we presumably would not want to invoke the law against the bias that doctors are male. In brief, there are many problems. To continue. These biases are in natural languages, and we are immersed in these languages. Likely there is some two-way traffic between the languages we use and the biases we have. Our linguistic practices do change over time— we, in English speaking America, are no longer comfortable with phrases like 'yellow peril' or words like 'ni**er'. That our linguistic biases have changed does not mean that shortly we will be free of all linguistic bias. Some fear that ML will amplify or entrench the existing biases in natural languages. It is hard to know. One factor that is awkward here is that much of NLP ML is unsupervised or self-supervised. That means that the programs are often not being told the 'right' answers. The language corpora that they work on are almost always huge. The GPT-x series, for example, essentially scan the entire Internet, maybe trillions of word tokens. If a program is looking for patterns, undirected, through the whole of the Internet, it is hard to see how it could omit biases from that search (it does not even 'know' that they are biases). (GPT-3 itself, for example, can be given prompts, which can give it some direction. Prompts can tell it to be safe and not to be toxic or biased.) Some biases can be reduced or removed in ML software on a piecemeal basis. There are de-biasing initiatives (e.g. (Bolukbasi et al. 2016)). We should take into account here considerations of offensive speech, hate speech, and legal protections of free speech (such as the First Amendment in the United States). Google, and similar large outlets of speech, have policies and guidelines on being parties to the

publication of hate or offensive speech. Basically, the policies respect the laws while keeping themselves clear of what might be marginal cases. The biases that occur everywhere in everyday language would not be front and center. Some tentative conclusions are… ML needs to use NLP to produce translations, sound interfaces, verbal assistants, and so forth. These technological possibilities are, on balance, so valuable that it is hard to imagine not pursuing them. Then data from NLP will likely contain bias, and that bias will be hard to address.

7.5 Some Clarification of the Term 'Algorithm'

Presumably, some ML programs are biased (just being open minded here on what the word 'bias' might mean in this context). But we need to be measured in addressing a serious issue. Here is what some prominent commentators write:

> Algorithms are neither neutral nor objective. They are programmed by human beings, who have both conscious and unconscious biases, and those biases are encoded into software (Cordell 2020).

> … essentially a lie— namely, that algorithms were being presented and marketed as objective fact. A much more accurate description of an algorithm is that it's an opinion embedded in math (O'Neil 2018).

> … algorithms are the result of human endeavor and human-generated data sets so they are just as biased as we are. We just can't see it.
> As humans, we all have implicit biases. And as we build these new systems – facial recognition, AI, analytical algorithms – we're

> creating them in our own image, with these biases baked in. (Ayre and Craner 2018).
>
> … algorithms are the product of explicit and latent biases held by humans (Padilla 2019).

The sentiments expressed here are both factually wrong and pernicious. The Ryan Cordell passage, to take one example, is from a report for the Library of Congress, which is the most important institutional body in American librarianship. The Library of Congress here thus presumably approves of, and certainly promulgates, a report that misleads librarians.

Most algorithms, computer science algorithms and folklore 'algorithms', are not biased. [See Section 1.7. For example, the algorithm division by successive subtraction is not biased, period.] Separately, the argument 'All humans are biased, therefore, all human products (e.g. software) are biased' is invalid and has a false premise. Some more detail, or evidence, can be added here. The US federal agency IMLS (Institute of Museum and Library Services) has funded a useful and informative educational resource on algorithmic awareness, aimed to an audience of information professionals (Clark [2018] 2022). The resource mentions, and demonstrates, the following as important algorithms, used in online search:

> … PageRank, merge sort and heap sort, Dijkstra's algorithm, link analysis, and TF-IDF (Term Frequency-Inverse Document Frequency) (Clark [2018] 2022).

None of these algorithms is biased. Take Dijkstra's algorithm, for example. As an analog of what it does: it will find the shortest path, or route, between

any two cities, where there are several cities connected by roads (sometimes a direct route is shortest, sometimes going via intermediary cities is the shortest). This algorithm is not biased (in any sense of 'bias' whatsoever). Here is a general argument to refute the view that all algorithms are biased: Dijkstra's algorithm is an algorithm, Dijkstra's algorithm is not biased, *ergo*, not all algorithms are biased.

We do not wish to get tied up here with arguing the meaning of words. If those concerned with shortcomings with the use of computers in society regularly talk about 'algorithmic bias', the 'social power of algorithms', or even '#FuckTheAlgorithm', that is fine (Beer 2017; Benjamin 2022). We will open our minds to this. For ourselves, we prefer 'bias in the programming artifacts', or 'bias in the algorithmic pipeline' or just the plain 'bias in the software'. We will cautiously and tentatively use this, and similar phrasing, in the case of ML algorithms. What we will not do is buy into the argument 'we are all biased, therefore all our algorithms and computer software are biased'.

7.6 Computer Program Inadequacy

Some computer programs are unreliable. That will not come as news to you. Some unreliable computer programs are used in circumstances where their output, advice, or decisions have serious human or material consequences. A regularly cited example is risk assessment in the criminal justice system (Tashea 2017; Angwin and Larson 2016; Budds, Budds, and Budds 2017). In some jurisdictions software is used to predict whether convicted felons are at risk of offending again. This kind of software can

make false positive errors in classification (that a felon is at risk of reoffending, when the felon actually is not) and false negative errors (that a felon is not at risk of reoffending, when the felon actually is). (See Appendix C.1 for further explanation of false positives and false negatives.) Certainly, some examples of this software seem to be very poor. There is also a more general concern in this setting and that is: in a court of law the purported evidence should be transparent and contestable. The parties should know what it is and be able to argue about it. But many ML systems can lack transparency and not be revealing about their inner mechanisms (Liu 2019; Abebe et al. 2022). There are also many other examples of poor, and potentially damaging, classification software (e.g. credit ratings, job application assessments, mortgage lending decisions) (O'Neil 2016).

Computer software does not have to be unreliable. Some software can be proven to be correct— there are mathematical proofs that the software meets its specification. Other programs can be validated and evidence, and certification, provided that they meet requirements. There is a considerable portion of software engineering given to correctness and assurances of performance in mission critical settings. The development of programming languages and programming techniques has been in part driven by the need to produce quality in the face of complexity. Any computer science graduate will have had exposure to questions of how to ensure that a program is correct and how to produce evidence that it is. Just to give a couple of examples:- there is Unit Testing, where test code is written at the same time as the actual programming and run automatically over and over as the program develops, and Extreme Programming, which emphasizes teamwork in the actual programming. That said, there still can be

unreliability in the end result, and, in the case of ML, there is another factor. Many of the programs are quasi-empirical.

ML and DL can often be more akin to empirical science than they are to traditional computer science and the practices of software engineering. What is being asserted here? In general, our knowledge can be divided into empirical knowledge and non-empirical knowledge. Empirical knowledge is knowledge assured by observation and experience. One form of empirical knowledge is that provided by science. Scientific method includes deliberate experimentation, random controlled trials, natural experiments, and the like. Science is conjectural and fallible— there is the permanent possibility of error. It is also, for the most part, implicitly or explicitly, probabilistic (Howson and Urbach 2006). The theories, explanations, and predictions involve probabilities. In contrast, non-empirical knowledge, for example, mathematics, is knowledge assured by logic and reason. It is not usually conjectural, fallible, or probabilistic.

Computer programs, their correctness, and our knowledge of what they do, are generally in the domain of the non-empirical. (There are exceptions such as non-deterministic algorithms, genetic algorithms, and so forth, but these are but small paddocks in the large continent of computer science.) ML and, especially, DL, is another matter altogether. It is on the empirical side of the divide— it is quasi-empirical— and, also, most of its predictions are probabilistic.

Many DL systems are black boxes— quite how they work internally in specific implementations is often unclear. Then whether they actually do

work as anticipated is often a matter of experiment and empirical test. Evidence is gathered by providing the systems with data and seeing if they behave as they should. The testing is made more complicated by the probabilities simply because any probability other than 0 or 1 is consistent with any actual outcome in the world. For example, if risk assessment software predicts that a specific felon has an 80% chance of re-offending, that prediction is *not* refuted by the felon *not* re-offending. (Just as, if the weatherman does not have to be wrong by saying that there is an 80% chance of rain and then, actually, in reality, it turns out that there is no rain on the day in question.) Science itself is empirical, and it has evolved techniques for dealing with the probabilities. So, DL is not beyond redemption here— it is just that there are challenges and deeply entrenched fallibility. Humbleness is the order of the day.

7.7 Bias in the Context of Wider Machine Learning Programs

Let us first consider what ML can and cannot do to address unfairness and representation in a general setting. This is important for librarians to know. Librarians offer education in 'information literacy'. Knowledge of the properties of ML will increasingly become a core aspect of this. Then we will look at ML unfairness, representation, and classification, specifically in the case of librarianship.

7.7.1 Fairness ('Distributive Justice')

> The economic, political, and social frameworks that each society has—its laws, institutions, policies, etc.—result in different distributions of benefits and burdens across members of the society. These frameworks are the result of human political processes and they constantly change both across societies and within societies over time. The structure of these frameworks is important because the distributions of benefits and burdens resulting from them fundamentally affect people's lives. Arguments about which frameworks and/or resulting distributions are morally preferable constitute the topic of distributive justice (Lamont and Favor 2017).

ML has little or nothing to add to the vast and supremely important topic, or concern, of being fair, the topic of distributive justice. ML is statistics concerning facts, it does not offer moral guidance.

However, ML can itself supply facts that allow decisions to be made. Also, research in ML has also produced some surprising results concerning fairness (e.g. some obvious strategies for being fair can harm the folk they are trying to be fair to).

We need some background to introduce ML into a discussion of being fair and unfair. Assume a classification is going to be done into two classes on the basis of a single numerical score (see, for example, (Wattenberg, Viégas, and Hardt 2022) and (Hardt, Price, and Srebro 2016)). These classes are used to make a prediction for entities that are being classified, and resulting prediction is either correct or incorrect. An example might be a judgement on whether a person will pay back a mortgage, using 'credit-worthiness' as the numerical score. We have information, or data, on the past scores, and

we also have information, or data, on whether the borrowers paid their mortgages back. The data may be like this:

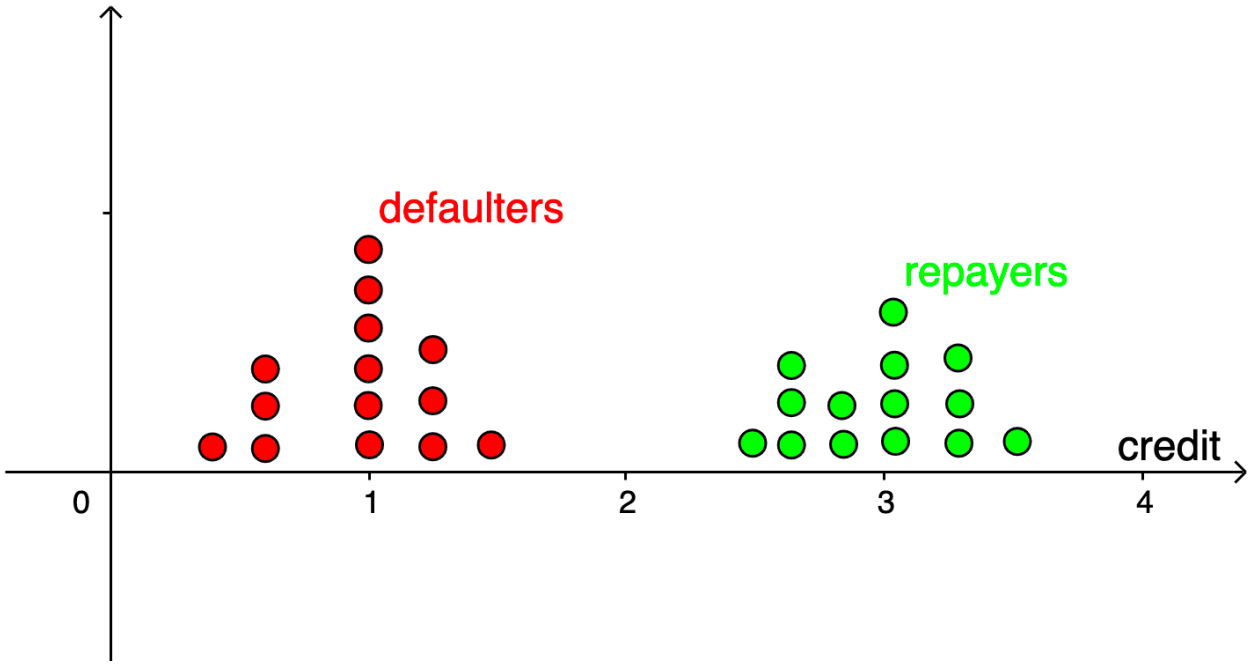


Figure 20. Graphic depicting defaulters and repayers against 'credit-worthiness'.

The intention here is to depict that anyone represented by a red dot (i.e. with a credit score between 0.4 and 1.5) failed to repay their mortgage, whereas everyone represented with a green dot (i.e. with a credit score between 2.4 and 3.6) did repay their mortgage. It is easy in a case like this to put in a credit score that classifies or divides the borrowers into defaulters and repayers— having a credit score of 2, or above, will do it

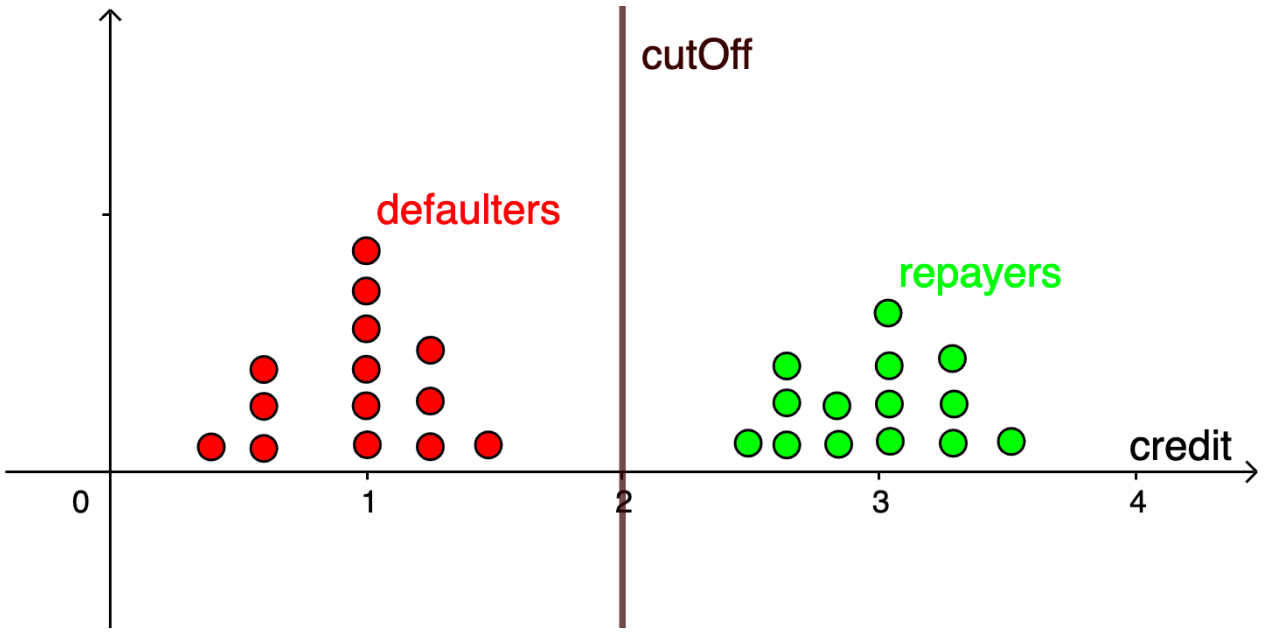


Figure 21. Graphic depicting defaulters and repayers against 'credit-worthiness' with a cut-off line.

This putting in of a cut-off line— a 'threshold classifier'— is a 'theory' that works perfectly on past data, and we will assume that it holds good with future data going forward. It is a theory that may have been devised by ML, or it may have been produced in many other ways. (After all, mortgage companies had similar theories long before the arrival of ML.) Problems with the theory start to arise if in the actual data these two regions overlap, say

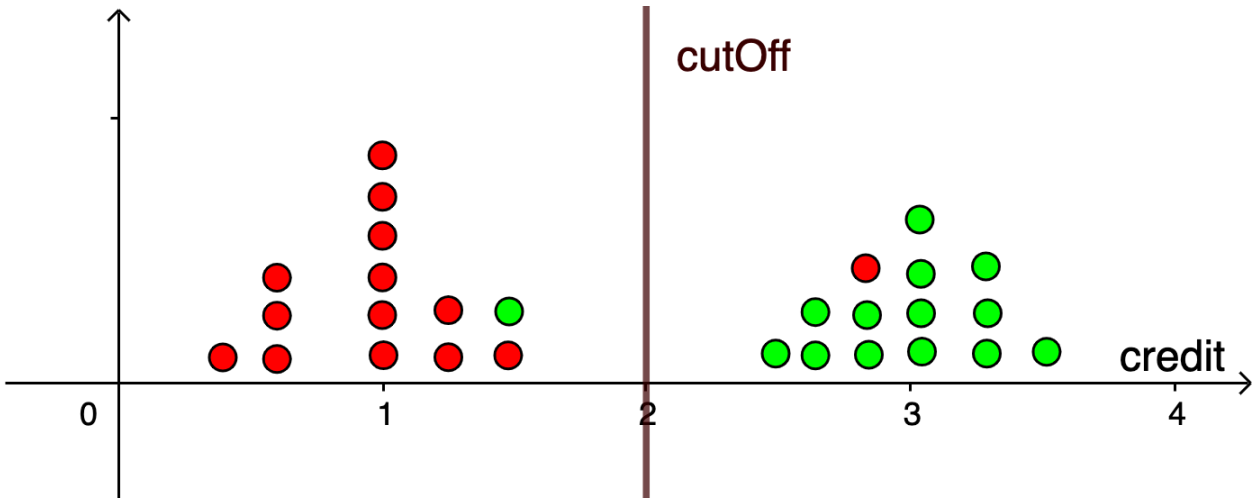


Figure 22. Graphic depicting defaulters and repayers against 'credit-worthiness', where there are false positives and false negatives.

With the data depicted in Figure 22, there is no way of putting in a classification boundary that does not make some errors. There are two kinds of errors that might be made in respect of repayment: false positives and false negatives. The false positives are where the classification suggests that borrower will repay, but the borrower does not. The false negatives are where the classification suggests that borrower will not repay, but, actually, the borrower does repay. For example, say a boundary was at credit 2.0, as in Figure 22, there is one false positive and one false negative. Of course, in a real case there might be thousands of borrowers, and hundreds of false positives and false negatives. No matter where the cut-off line is put there will be errors of these kinds. It is inevitable with this data. Statistics has techniques for adjusting theories to minimize errors. We will not invoke those here. Instead, we will leave this part of the discussion noting that

likely there will be false positives and false negatives. (Please see Appendix C for a further explanation of false positives and false negatives.)

That there are going to be errors is not a good result. Maybe the composition of the single score could be improved so as to separate the classes. This single number credit-worthiness score would typically be an amalgam of many other numbers, i.e. features, for example, salary, number of years of employment, family size, etc., adjusted with weights to reflect their importance. A DL approach would typically add features (any features not required would just end up with weight zero). This might help. But there will always be other general worries about the training data, whether the sampling properly represents the target population, whether the data on the features and the predictions (the labels) are correct, and so forth. A prudent conclusion might be that there will always be errors.

There is another aspect to what we know and what we do not know here. We know only probabilities. We do not know of a particular future borrower whether that individual borrower will repay. We know, for example, of borrowers like that borrower (i.e. ones with the same credit score) that, say, there is a probability of 90% that they will repay.

Let us move on from a factual judgement (on who is going to repay) to policy. Just as far as the actual lending goes, the mortgage company does not have to act inexorably on the advice or prediction of its repayment theory. It may have other reasons for lending or not lending. For example, it may have only so many funds to disburse and thus be forced into not lending to some clients whom the company knows would be perfectly good

repayers (or would have a very high probability of repaying). Let us now understand the cut-off in a slightly different way, as a major input to policy regarding future applications— that it is a main factor in separating those suitable for a loan from those not suitable.

The boundary of suitability can be adjusted to alter the proportions of false positive 'suitables' or false negative 'unsuitables'. For example, the mortgage company could potentially loan to every applicant (i.e. the cut-off would be at a credit score of zero); then there would be no false negatives (but, presumably, a number of false positives). Or, it could move the cut-off far to the right and have no false positives (but a number of false negatives). The mortgage company has choices. There are going to be errors. But the kinds and numbers of errors can be manipulated.

Now let us introduce fairness. Among the features in the single number credit-worthiness score there may be values for attributes, for categories, that would merit scrutiny in the context of bias and fairness, for example categories like gender and race. In this area of research, these categories are known as *protected* or *sensitive* features or categories.

Let us introduce a fictitious sensitive category, Shape, which has two values {circle, cross}. We can show these on our diagram.

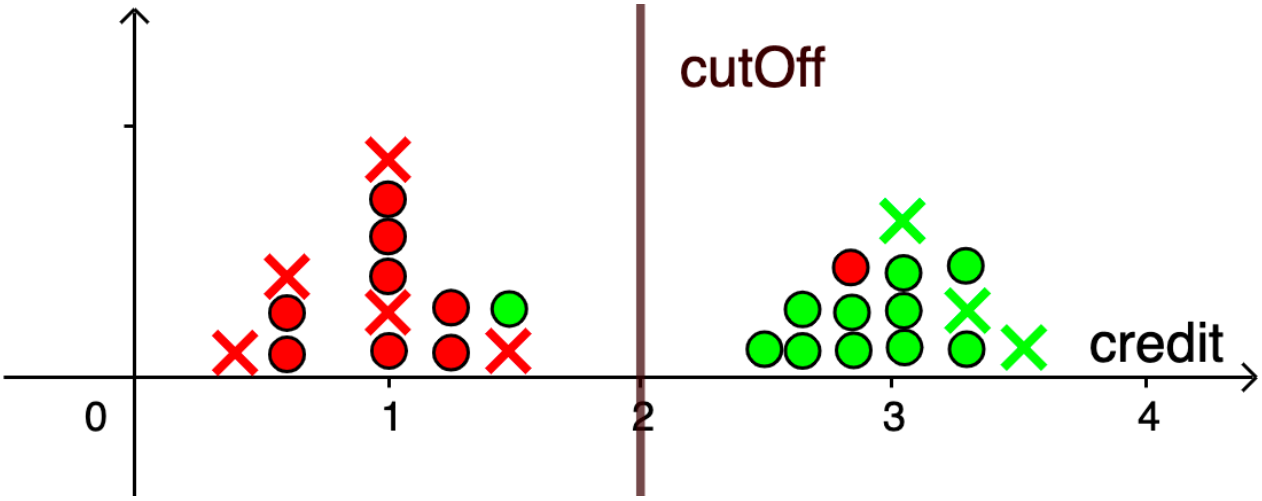


Figure 23. Graphic depicting defaulters and repayers against 'credit-worthiness', where there are false positives and false negatives and protected features.

Is our theory unfair on Shape? What might 'bias' mean in this context? How might we counteract bias? For the moment let us set aside the question of whether shape, e.g. being a cross, may cause a borrower to be a better or worse repayer. We will come back to that.

Here are some suggestions (Hardt, Price, and Srebro 2016; Corbett-Davies and Goel 2018; Kusner et al. 2018)

> Approach 1 ('unawareness' or 'anti-classification'): Ignore the property of Shape. The suggestion is that it is unbiased, or not unfair, over Shape, in as much as it ignores Shape. Analogically, in a more realistic setting, the suggestion is that one way to avoid bias over Race and Gender is not to have, or not use, any data on race and gender. There are two problems with this, though. Shape (or Race or Gender) may be correlated with other features that have to potential to serve as proxies for the protected attributes. For example, maybe all the crosses live in one zip code, and the

circles in another zip code. Then a strategy of ignoring shape, but using zip code, may lead to, or reveal, unfairness on Shape even though the algorithm does not directly use data on Shape. It is better to have data on Shape and to prove that the algorithm produces fair results (using an acceptable definition of fairness and a technique for achieving it.) The second problem is that the protected attributes may indeed have a causal relation to the predicted outcome label. It may be that the crosses are better repayers than the circles, so omitting information on this may produce a theory, a cut-off, that is less accurate (leads to more false positives and false negatives). This may seem unlikely or implausible in a mortgage repayment example. But in a medical setting, it is known that there are many differences between the races and genders. Whites, in the US, are more prone to certain heart conditions (e.g. atrial fibrillation) than other races (Dewland et al. 2013). Sickle-cell anemia is predominantly a disease of those who live in sub-Saharan Africa (and descendants of earlier residents of that region) (Rees, Williams, and Gladwin 2010). Women, and not men, have reproductive systems for bearing children. Men, and not women, are exposed to the possibility of prostate cancer. Women live longer than men. It seems that information on race and gender would be useful in medical settings. The systems would need to be 'fair' in their uses of that information, but not using the information at all does not seem to be the right move. Ignoring sensitive attributes can lead to unfairness with those the system is trying to be fair to. Female violent felons have a lower rate of recidivism than do male violent felons. Omitting gender from recidivism calculations may lead to harming women.

Approach 2 ('demographic parity'): separate the data into two sets of data (two graphs)— one for circles, the other for crosses— then, potentially, use a different cut-off for each ensuring that the same proportion are candidates for loans. So, for example, if there are a thousand crosses, and the cut-off for crosses leads to 10% of them qualifying for loan, then set the cut-off for circles to ensure that 10% of them qualify (whether there be 100 circles or 10,000 circles). Some defense can be made of this, in certain circumstances. But consider the true negatives, on either graph. It may be that a mortgage company, following the demographic policy, has to lend to borrowers that they know will not repay. At

an extreme, say all the crosses repay and none of the circles do (i.e. repayment is causally related to a protected attribute). If 10% of the crosses are offered loans, then the demographic policy requires that 10% of the circles also be offered loans, even though they are not going to repay. This actually does not seem fair to anybody (or to the company).

Approach 3 ('equal opportunity'): again, separate the data into two sets of data into two graphs— one for circles, the other for crosses. And, again, there will be two cut-offs. But this time the cut-offs focus only on those who are judged to be repayers. Then, the cut-offs are set to ensure, in so far as it is possible, that the same proportion of circle repayers and the cross repayers are offered loans. If you are classed as a repayer, there is equal opportunity of being offered a loan, whether you are a cross or a circle. (Wattenberg, Viégas, and Hardt 2022) and (Hardt, Price, and Srebro 2016) favor this approach. Presumably there is the problem with it of false negatives. Say you are a repaying cross. Some of those are going to be incorrectly classified as non-repayers (i.e. they are false negatives). But once they are (wrongly) classified as a non-repayer they will not have an equal opportunity of anything. They are not going to be offered a loan and nor do they have a chance of being offered a loan. It may be that equal opportunity is fair for the group but not necessarily fair for every member of the group individually.

Approach 4 ('counterfactual fairness'): (Kusner et al. 2018) suggest the following. Work with individuals only. Then require, and prove, that the probability of getting a loan for any individual, who is actually a cross, is exactly the same as the probability of that same individual getting a loan, had that individual been a circle (i.e. counterfactually being a circle) and similarly in the other direction, from circles to crosses. That is, data on sensitive attributes is obtained and used. But it is used to show that the outcome results would be the same for all individuals even were the values of those sensitive attributes were different. This approach certainly plumbs a central intuition. For example, under it, with race, whether you are black or white does not matter as far as your probability of getting a loan is concerned. While counterfactual fairness is different conceptually to demographic parity, Rosenblatt and Witter have proved that the two lead to

equivalent outcomes (L. Rosenblatt and Witter 2022). Demographic parity is easier to work with.

To sum up. Fairness in ML algorithms is an active research area. There are a number of proposals. Most of them have merits and shortcomings. Since there are probabilities involved, with false positives and false negatives, it seems unlikely that any suggestion on fairness in ML, can, at one and the same time, be fair to all individuals, to all groups, and to all related parties. Most of the proposals can be proved mathematically to hold, or not hold, of the relevant ML systems. There can be evidence and accountability. Which theory of fairness should be used is not a matter for the ML programmers to decide. It belongs with distributive justice, and it is a decision for the wider constituents.

There is a take home here. If there is a test that has false positives and false negatives (and pretty much all real-world tests do— medical tests, driving license tests, law school admissions tests, etc.). And if some, many, or most of the false positives have some other property (say, having the race of 'green', or 'being a cross'). Or if some, many, or most of the false negatives have some other property besides testing negative (say, having the race of 'red', or 'being a circle'). None of that, by itself, means that there is any evidence whatsoever of unfairness. Further analysis is needed— further statistics, or further mathematics. Epidemiologists— one group with a knowledge and interest in these methodologies— have tools at their disposal. One is causal diagrams. The use of causal diagrams, with appropriate data, can provide (fallible) evidence for fairness or unfairness. [Causal diagrams are explained and discussed further in Appendix D.]

7.7.2 Debiasing Representation

> Man is to Computer Programmer as Woman is to Homemaker? (Bolukbasi et al. 2016)

That attention grabbing question, or phrase, is part of the title of an important paper by Tolga Bolukbasi and fellow authors. What they are alluding to is that natural languages have associations in them which reveal assumptions about gender stereotypes. Sometimes these assumptions are innocent, harmless, and possibly even useful, such as the association between being a Queen and being female. Often, though, associations between words can be suspect and perhaps even revealing of undesirable underlying biases, such as that between 'receptionist' and 'female'. A problem in the context of ML is that if ML uses natural language as data, and it often does, the resulting ML programs might entrench or even amplify the biases. While this Bolukbasi paper focusses on gender stereotypes, it also would have application with racial or religious or other stereotypes. (See also (Caliskan, Bryson, and Narayanan 2017).)

This type of bias is different to the unfairness biases of allocation, for example, as to who gets mortgages and who does not. Rather, this is to do with biases of representation, with natural language processing (NLP) and with removing unwelcome stereotypical associations. Natural languages change, of course, and unwelcome stereotypical associations come and go. How to interact with that in a positive way is a larger question. But reducing bias, or 'debiasing', text which is used as input data to ML is a distinct possibility. The Bolukbasi paper has definite sound proposals on

this. It is not being asserted here that text for ML can be 'purified' perfectly. However, the text can be improved, and it should be possible to ensure that ML programs do not amplify existing biases in language.

There can be other harms of representation in addition to those strictly in NLP. For example, there can be such harms with the labeling of images—the attachment of metadata to images. In the case of the single sentence characterization of an image there might be denying people the opportunity to self-identify, reifying social groups, stereotyping, erasing, and demeaning (and probably further types) (Wang et al. 2022).

7.7.3 Panopticon Bias, the Panopticon Gaze

Certainly computers, artificial intelligence, and ML are enabling surveillance as never before. Examples of this are readily available in librarianship. There are recommender systems which can recommend books, articles, music, films, etc. that individual patrons might like. But to do this, the systems have to know what at least some patrons have read or explored in the past. Likely, patrons will have to give up some privacy to get the value-added intermediation of recommender systems. A more extreme example is that facial recognition software could track everything that every patron does in a physical library. This would be completely against the ethos of librarianship.

Facial recognition technology certain raises questions. It is a technology that allows the identification and tracking of individuals. These days it is

pretty good in a technical sense i.e. good at identifying and tracking. But, Nick Thieme asserts:

> AI's unique talent for finding patterns has only perpetuated our legal system's history of discrimination… Since people of color are more likely to be stopped by police, more likely to be convicted by juries, and more likely to receive long sentences from human judges, the shared features identified are often race or proxies for race. Here, computational injustice codifies social injustice. (Thieme 2018)

Joy Buolamwini has written on topics related to this. One of her early papers observes that she— a person of color— was largely invisible to computer systems, then later she offers the view that computer facial recognition was a technology of discrimination against people of color (Buolamwini 2019; 2016; *Race, Technology, and Algorithmic Bias* 2019). She has a new 2023 book *Unmasking AI: My Mission to Protect What is Human in the World of Machines* (Buolamwini 2023) The American Library Association also have a piece, now mildly dated (American Library Association 2018). We can all agree that recognition and tracking is a creepy technology that seemingly we can do without.

Or can we? There are many occasions when there is a need to know a person's identity— i.e. who the person is. In librarianship, there is the need to know who the patron is that is checking out the books. To establish identity there needs to be some gold standard, some difficult to forge validator whose original is on file or permanent record somewhere. Biometrics offers a way in here: it can use images of faces, fingerprints, images of irises, DNA, and similar. Right now, facial images are by far the

best combination of ubiquity and convenience. More-or-less everyone in the US has ID (identification) and that ID is going to be a driving license, a Real ID equivalent, a passport, a Green Card, or similar. All of these carry an image, a photo, identifying the holder of the ID. We can add to this folk who unlock their smartphones using a scan of their face. Facial recognition itself is now so good that it can recognize a person, in person, from a suitable image with 99% or more accuracy (quite what this 99% figure means is another question). Let us insert an anecdote. In June, 2023, the author flew from Dallas to Paris on American Airlines. When boarding he walked straight on to the aircraft in seconds, being identified by facial recognition (the airline, along with many others, already had a scan of his passport). Now, this facial recognition presumably was being trialed and not mandatory. But, also presumably, objectors would have had to produce their passports, to have printed and produced their boarding passes, and to have spent minutes with these processes. There will be no need to make facial recognition mandatory for these kinds of circumstances. We will all want it, for convenience. We will be falling over ourselves to get it. [Hot off the press, *The Independent* headline 7/18/2023 'Eurostar passengers leaving London can skip passport checks with new facial recognition tech'.] Separately, more than a few sporting venues use facial recognition technology to identify season ticket holders and to admit them without fuss or muss (Gee 2023). The US Immigration and Naturalization Service use facial recognition at airports to identify persons of interest. The author has been through INS at US airports many times. Every time until recently the INS agent took his fingerprints. On coming back from Paris in 2023 this did not happen. The agent told him that it was no longer necessary. Who knows why? INS must have had all the identification they needed. Perhaps from

facial recognition? It is hard to see all this being rolled back. Facial recognition has uses which are absolute winners. Of course, tracking people 24 hours a day, 7 days a week is an entirely different matter. There are companies (e.g. IBM) who say they will not sell this technology to the police (Peters 2020). There is an important difference between being recognized through a driving license in a pocket and being identified by facial recognition. Ordinarily, an officer of the law, or similar, would have to ask to see a driving license in a pocket, and the person asked might consent or refuse to reveal it. Then this kind of transaction would not scale, say to 10,000 people in a crowd at a protest. Facial recognition, though, can work with or without assent and it scales easily to many thousands of faces.

7.7.4 Bias in (Librarianship) Classification

This topic is included in Chapter 7 Machine Learning Bias and Librarianship.

7.8 Stochastic Psittacosis: LLMs and Foundation Models

The three reports or papers *On the Opportunities and Risks of Foundation Models* (Bommasani et al. 2022), *Language Models are Few-Shot Learners* (Brown et al. 2020), and *On the Dangers of Stochastic Parrots: Can Language Models Be Too Big?* (Bender et al. 2021) have a wealth of material on potential harms arising from Large Language Models or Foundation Models. We need to be aware of some of the problems.

Large Language Models are large, no surprises there. Then what they do, essentially is cloze tasks (see Section 3.5, for an explanation of cloze tasks). This can lead to many other abilities (question answering, chatting, reasoning etc.). Nevertheless, what seems to be happening is probabilistic symbol manipulation on a grand scale. There is no doubt that some of these systems would pass the Turing Test (which at one point in time was taken to be an indicator of whether a system had intelligence (Oppy and Dowe 2021)). Nowadays received opinion is that the Turing Test is not demanding enough. There is the open question of whether there is emergence here. One view is that as the cloze tests, and the models mastering them, get ever more elaborate, something 'emerges' from the complexity, and perhaps that emergent property is true intelligence or even consciousness or sentience. A contrary view is that we can easily fool ourselves over apparently intelligent behavior. Maybe even 50% of the population would think that the ELIZA chatbot is either a real person or some truly intelligent software. The contrarians would just view the LLMs as being more complex symbol manipulators. This debate matters in the following way. We need to be cautious and skeptical towards what these modern sophisticated models have to say and recommend. We do not really know how they work in detail. Nor do we know, or can explain, the reasoning that produces many of the specific results. They have characteristics of a black box or oracle. Viewing them as stochastic parrots— to use (Bender et al. 2021)'s delightful label— lessens our awe in what they seem able to do.

One potential harm is misuse. GPT-3 can write, say English, as well as a native educated speaker. This opens some unwelcome possibilities:

> Any socially harmful activity that relies on generating text could be augmented by powerful language models. Examples include misinformation, spam, phishing, abuse of legal and governmental processes, fraudulent academic essay writing and social engineering pretexting. Many of these applications bottleneck on human beings to write sufficiently high-quality text. Language models that produce high quality text generation could lower existing barriers to carrying out these activities and increase their efficacy (Brown et al. 2020).

Basically, there is very little that can be done about this. It may become possible for other AI applications to recognize, and maybe filter, machine written text. If the writer application always puts in a 'watermark' (some giveaway combinations or sequences of words), it might be easy to recognize AI generated text. Using a machine to generate text is not always bad. For example, summaries of text, journal articles, and even entire document collections, can be exactly what readers require.

Another potential harm is bias. Bias is a huge and complex topic (as we have been seeing). Just briefly here we will use Kate Crawford's distinction between harms of allocation, harms of representation, and harms of classification (Crawford 2017). Harms of allocation, e.g. fairness in the availability of mortgages, should be able to be addressed. Within the bounds of sloth and fallibility, unfairness can be detected and remedied. Harms of representation, e.g. Muslim-violence bias (Abid, Farooqi, and Zou 2021), are a much harder case. There are hundreds of training data sets of (English) samples. The ones of these that are large and containing substantial source material from the Internet (e.g. from Common Crawl) will have some biased content. Then, if the training data has bias then so

too will GPT-4 and similar large models. To address this, there seem to be two central possibilities: to keep the unacceptable bias out of the training data, or to remove the bias from the model's output. Neither of these is promising. The systems use self-supervision on the training data precisely because it is near impossible to curate and label with the quantity involved. It may be that some software could filter the training data in some way. But that might not be entirely suitable. GPT-4, if it is going to have an all-around knowledge, needs to know about biased content (for example, holocaust denial). It 'just' needs to know that the biased content is biased and to not write biased content when writing in its own voice. Working on the output probably is not much better. Our experience with filters (e.g. using blacklists) on the Internet is not good. Filtering the word 'breasts', for example, to filter pornography tends also to filter 'breasts' in the context of breast cancer. It may be possible with systems like GPT-4 to instruct the systems themselves to remove biases of representation. For example, to provide examples of, say, anti-Muslim bias and prompt the machine to remove material like this from its reasoning and output. Harms of classification also are tricky and require attention. The actual classification categories used to classify people for some purpose, for example, usually or often depend on historical period, culture, and social factors (Hacking 1999). But training data for a foundation model will usually favor one time, place, and culture. This issue is also seen with medical data. There are many labeled sets of medical data. But many of them would not be good as training data. One reason is classification categories and diagnoses can change through time. Views of homosexuality in the 1950s, for example, are different from those of today.

7.9 Supplement: The Bias of Programmers

7.9.1 The 'Biases' of Professional Programmers

> … [the] possibility of programmer biases being encoded, in some way, in their programming artifacts.
> Contrary to making an error, which represents a single incident in which one makes an incorrect judgment, *a bias* is a systematic tendency to commit the same type of error over time or in different situations (Johansen, Pedersen, and Johansen 2021).

This kind of bias has absolutely nothing to do with bias in the sense of being unfair to anyone over race, gender, social status, economic status, etc.

A typical workflow process for a professional programmer at work on an individual program is that there will be a specification that the program must meet. There will be a 'house style', which is how the programs are to be written, for the employer, or for the open-source project, or for the intellectual area in question. When the program is first completed, and continuously thereafter, it will be subjected to quality assurance (including 'debugging'). The program will be shown to meet the specification. The programmer will have learned programming in MIT, Stanford, Princeton, Lomonosov State University in Moscow, or in many other worthy institutions including polytechnics and community colleges. They may even have learned from online sources or have been self-taught. They will have learned a *style* of programming. Not every programmer is different from every other programmer— definitely there are styles among programmers.

These are not 'biases'. All programmers make errors, but they will catch most of these in the debugging (and if they cannot, likely the program will not meet the specification). Most of the types of bugs are known. For example, there is an 'off-by-one-error' in a loop. (We need not worry about what this is.) Even the best programmers can make an off-by-one-error, but if a programmer has the systematic tendency to make off-by-one-errors (i.e. they are biased over this), basically they need to find a new line of work.

Johanna Johansen et al. argue that:

> … each program *probably* encodes the cultural and cognitive biases of their creators (Johansen, Pedersen, and Johansen 2021). [Emphasis added.]

and they offer evidence to this end. They even offer evidence of priming (that is, they can 'prime' a programmer to exhibit specific cultural and cognitive biases). Theirs is important work. But let us look at how it proceeds. There is a programming task— to write or sketch a program— but they omit any real specification for what the program is supposed to do. In this way they place the programmer in a condition of uncertainty. Then, their method suggests, the programmers reveal their cultural and cognitive biases in their decisions on what the program is to do, what the specification might be, (and this can be primed). While this research is important and addresses a rarely researched area, we find it unconvincing. Your mileage may vary. But our mileage is— if there are no specifications, all bets are off. There are starting to be other research papers in this area, and Johanna Johansen et al. provide a valuable guide to these (Johansen, Pedersen, and Johansen 2021).

7.9.2 The Biases of All of Us as Programmers

Pretty much all of us are programmers. There are six and a half billion owners of smartphones in the world ((Turner 2018) updated to 2022). That is over 80% of the world's population. Those phones likely have applications ('apps') on them, and settings. They are configured. That configuration, in conjunction with the host operating system and infrastructure, amount to a suite of programs, a cluster of software and software infrastructure. The owners of the phones, or their friends, or (young) relatives, or delegates will establish or program the configurations. There is a lot of programming going on. There will be extensive bias in the sense of systematic mistakes in the programming, and also the phone will enable bias, prejudice, in the sense of bringing to light unfair views or opinions about other human beings.

The importance of this is that to a large degree how our own smartphones, and similar, are set up are within our own control. We can reduce bias if we wish to. What we need is 'information literacy' and configuration agents to implement the prescriptions.

7.10 Annotated Readings for Chapter 7

Algorithmic Justice League. “Algorithmic Justice League - Unmasking AI Harms and Biases,” 2022. https://www.ajl.org/. (Algorithmic Justice League 2022)

American Library Association. “Facial Recognition.” Text. Tools, Publications & Resources, 2018. https://www.ala.org/tools/future/trends/facialrecognition. (American Library Association 2018)

Angwin, Julia, Jeff Larson, Surya Mattu, and Lauren Kirchner. 2016. “Machine Bias.” Text/html. ProPublica. 2016. https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing. (Angwin and Larson 2016). This criticizes risk-assessment software in the case of Northpointe's software and Broward County, Florida. The article brings the area to life with real aberrant cases which presumably amount to false positives or false negatives (and which involve the sensitive feature of race). It links to a description of its own methodology and full data set. Northpointe do not agree with the analysis and its findings.

Bolukbasi, Tolga, Kai-Wei Chang, James Zou, Venkatesh Saligrama, and Adam Kalai. “Man Is to Computer Programmer as Woman Is to Homemaker? Debiasing Word Embeddings.” arXiv, 2016. https://doi.org/10.48550/arXiv.1607.06520. (Bolukbasi et al. 2016)

Caliskan, Aylin. 2021. “Detecting and Mitigating Bias in Natural Language Processing.” Brookings (blog). May 10, 2021. https://www.brookings.edu/research/detecting-and-mitigating-bias-in-natural-language-processing/. (Caliskan 2021). Beautifully clear and comprehensive, and at about the right intellectual level for us.

Glusac, Elaine. “Your Face Is, or Will Be, Your Boarding Pass.” The New York Times, December 7, 2021, sec. Travel. https://www.nytimes.com/2021/12/07/travel/biometrics-airports-security.html. (Glusac 2021)

HAI. “AI Index Report 2023 – Artificial Intelligence Index,” 2023. https://aiindex.stanford.edu/report/. (HAI 2023). This is 386 pages long. You could try reading Top 10 Takeaways, which is 2 pages long.

Katell, Michael, Meg Young, Bernease Herman, Dharma Dailey, Aaron Tam, Vivian Guetler, Corinne Binz, Daniella Raz, and P. M. Krafft. “An Algorithmic Equity Toolkit for Technology Audits by Community Advocates and Activists.” arXiv, 2019. https://doi.org/10.48550/arXiv.1912.02943. (Katell et al. 2019)

Rainie, Lee, and Janna Anderson. “Code-Dependent: Pros and Cons of the Algorithm Age.” Pew Research Center: Internet, Science & Tech (blog), 2017. https://www.pewresearch.org/internet/2017/02/08/code-dependent-pros-and-cons-of-the-algorithm-age/. (Rainie and Anderson 2017). Much of the material in this Chapter has been on the 'Cons' of algorithms. But there are also some 'Pros'.

Wikipedia. “Algorithmic Bias.” In Wikipedia, 2022. https://en.wikipedia.org/w/index.php?title=Algorithmic_bias. (Wikipedia

2022b). This article, as it was 9/29/2022, is extremely good— both broad and deep. All the topics and discussion are important. In the present text, we are just not keen to apply the label 'algorithmic bias' to them. The article itself says 'In many cases, even within a single website or application, there is no single "algorithm" to examine, but a network of many interrelated programs and data inputs, even between users of the same service.' That captures where we are coming from.

Chapter 8: Bias in Machine Learning and Librarianship

8.1 Introduction

Let us start this Chapter by revisiting the paper by Su Lin Blodgett et al. on bias in NLP (Blodgett et al. 2020). The abstract to this, in full, is:

> We survey 146 papers analyzing "bias" in NLP systems, finding that their motivations are often vague, inconsistent, and lacking in normative reasoning, despite the fact that analyzing "bias" is an inherently normative process. We further find that these papers' proposed quantitative techniques for measuring or mitigating "bias" are poorly matched to their motivations and do not engage with the relevant literature outside of NLP. Based on these findings, we describe the beginnings of a path forward by proposing three recommendations that should guide work analyzing "bias" in NLP systems. These recommendations rest on a greater recognition of the relationships between language and social hierarchies, encouraging researchers and practitioners to articulate their conceptualizations of "bias"—i.e., what kinds of system behaviors are harmful, in what ways, to whom, and why, as well as the normative reasoning underlying these statements—and to center work around the lived experiences of members of communities affected by NLP systems, while interrogating and reimagining the power relations between technologists and such communities (Blodgett et al. 2020).

This paper is very thorough in its reasoning, evidence, and citations. Please read it. We will take from it its three recommendations, applying them to librarianship:

1. Recognize the relationships between language and social hierarchies.
2. Encourage researchers and practitioners to articulate their conceptualizations of 'bias'—i.e., what kinds of system behaviors are harmful, in what ways, to whom, and why, as well as the normative reasoning underlying these statements.
3. Center work around the lived experiences of members of communities affected by [the] systems.

Some material for the first recommendation is in (Blodgett et al. 2020) itself. The third recommendation requires extensive outside empirical research which we are not equipped for. That leaves as our focus:

> ...what kinds of system behaviors are harmful, in what ways, to whom, and why, as well as the normative reasoning underlying these statements.

There is a vast literature on bias in information provision (and, of course, information provision includes librarianship). But much of this literature has a wider ambit than librarianship. It wants, for example, to 'interrogate' internet companies, and power structures of one kind or another in society as a whole. No comment is passed on that here. We need focus on ML. There is an intersection of ML, bias, and librarianship, which we will explore shortly. But, as a summary of what is to come. There is a problem with data. ML needs data to learn from. But at least some potential data from traditional librarianship might have unwelcome aspects to it. For example, an ML program could easily learn the cataloging task of applying *Library of Congress Subject Headings* (LCSH) to new books and resources. But some fear that the LCSH labels themselves are suspect. So, according to

some, bringing ML into it would just be to 'reify White Supremacy' (Cordell 2020).

8.2 Harms of Omission

In Christianity, no doubt among other places, there is the distinction between sins of commission and sins of omission— the first is doing something that should not be done, the second is failing to do something that should be done. Similarly with harms associated with ML: some are harms of commission, others harms of omission. When Crawford et al. discuss bias in terms of harms of allocation, harms of representation, and harms of classification (surveillance), and Suresh and Guttag extend this out to 7 harms, these are all harms of commission (Crawford 2017; Suresh and Guttag 2021). But information providers, including librarians also have an interest in something different. Provision is an aim, or to use a word with a wider span: 'service'. Failure to provide service can be a harm. It would be a harm of omission. Harms of omission are a little harder to deal with at a methodological level than harms of commission. With a harm of commission, what or who did it— the causal agent— is available. Whereas with omission, what or who did not do it— the absent causal agent— often needs to be identified and may not be identifiable.

8.3 What to Digitize

As mentioned in Section 1.4, ML algorithms need input data in the form of computer digital text i.e. as structured sets of 0s and 1s. For text sources, or text corpuses, that are not born digital, this raises the question of what to

digitize. In some areas, the practice certainly has been to digitize primarily what might be characterized as being 'white', 'colonial' resources: White libraries, newspaper collections of White newspapers, and so forth. This is a tricky area. We know from the fate of the Google Book project that many do not want their resources digitized, or even the resources of others (see Chapter 1.2). Also, we know that many peoples, tribes, or indigenous peoples, do not want some of their cultural artefacts recorded at all, let alone digitized. In contrast, there is the argument that digitization selection can be an anti-racist action (see, for example, S.L. Zeigler, *Digitization Selection Criteria as Anti-Racist Action* (Ziegler 2019)).

Perhaps the ML research and practical initiatives can stay on the sidelines in this debate. As Elizabeth Lorang et.al. write, concerning the Library of Congress's role:

> ... the technology itself will not be the hardest part of this work. The hardest part will be the myriad challenges to undertaking this work in ways that are socially and culturally responsible, while also upholding responsibility to make the Library of Congress's materials available in timely and accessible ways (Lorang et al. 2020).

8.4 Search, Primarily Using Search Engines

Search is a filter. The searcher is initially faced with some pages, a site, a collection of sites, or the entire Internet, then search reduces the rich vista to something suitable for the occasion. The vista is filtered.

There are different possibilities here, and different possibilities for distortion or bias. If the Search algorithm uses keywords, spelling correction, semantic correction, stemming etc., various mistakes or manipulations can occur. Louis Rosenfeld et.al. report that an early instantiation of a search engine on Amazon responded to searches for the subject ‘abortion’ with the question ‘do you mean ‘adoption’?' (Rosenfeld, Morville, and Arango 2015). This suggestion is rather more than spelling correction. It is also regularly reported that various configurations of search engines misdirect searchers for 'abortion providers' to 'adoption agencies'. Search sometimes works via recommender techniques comparing the searcher and the searcher’s task to similar searches by other patrons. What happens here depends on which groups the search is compared with, and this can be, in some sense, fair or biased. Usually, a search returns a list of links in order of relevance. Now, relevance is topic, person, and occasion dependent (Frické 2012). It depends on the keywords, the person searching (different people may get different results from the same keywords, and the occasion (the same person may get different results from the same keywords on different occasions). The latter two features or aspects depend on the degree to which the engine is tracking the User (and the engine does need to be aware of previous searches in order to disambiguate, narrow, and help the User). There is also manipulation of various kinds. For example, there is Search Engine Optimization (SEO) which is tricking the engine algorithms to place some urls or links higher than they otherwise would be. It is known that most Users will not look beyond the first few links that are returned from a search. This might not matter for the supply of 'pure' information. But, for example, if you were a commercial entity, you would prefer to have the links to your products within those first few. The

provider of the links prefers this, not necessarily the User. There are companies that provide those services. The returns may thus be affected by paid interests, such as advertisers, retailers, or political groups (although most engines will identify paid links). The search engine companies try to identify the techniques of SEO and to immunize or neutralize them. It is a continuing battle. But even among algorithms that rank by genuine 'merit', there can be different, and sometimes equally acceptable but orthogonal, views of merit. [Orthogonality here means this. Consider the example of sports cars: is the one that goes faster better than the one whose looks are more head turning? Or vice-versa? It is hard to know. The two properties are orthogonal— they are independent one with the other.] Search is a filter infused with judgments and values (and orthogonality).

Some areas of the Internet are cesspits. That is one side effect of ease of content creation and freedom of speech. Safiya Noble, in her book *Algorithms of Oppression: How Search Engines Reinforce Racism,* reports that her 2009 search engine query for 'Black girls' returned the porn site 'HotBlackPussy.com' as its first hit. Later studies— 2011 onward— produced similar results for the search 'Black girls' (and also for searches for 'Asian girls', 'Asian Indian girls', 'Latina girls', 'White girls'). One conclusion that Noble draws is:

> ... girls' identities are commercialized, sexualized, or made curiosities within the gaze of the search engine. Women and girls do not fare well in Google Search—that is evident (Noble 2018).

These specific results have changed out of all recognition now, 2023 (possibly in part as a result of Noble's research).

What a search engine returns for a query does not depend solely on the query. It depends also on who is asking the query— that is what personalization does— and the occasion and history of the question being asked. Having the query 'black girls' (and similar) answered with links including porn sites certainly gets our attention. But what specifically is the harm here and who is being harmed? Some of the folks asking this kind of search question might be offended by this kind of answer (and some might not be, and some may even by pleased by it). Working to correct offense is tricky, which is not to say that it is not worth doing. The harm seems to be elsewhere, with girls, in general, and of having their identities commercialized, sexualized, or made curiosities. It is a harm of representation. Search engines have made adjustments in this area.

Some years ago, when the world wide web was just starting, there were curated pages of links on topics. These pages were authored by humans. For example, Lycos and Yahoo did this. These curated pages are like bibliographies, even annotated bibliographies, that a librarian might produce. They did not have or use keywords as lead-ins. Had there been a curated page on 'Black girls' it may well have been neutral as to range of topics, in so far as that is possible. Nowadays, there are just too many pages and too many links to make this practical. Certainly, there are 'pillar pages', or 'topic pages', or 'topic clusters' (see, for example, (Clariant Creative Agency 2022)). These are somewhat similar to the old, curated pages, but likely they will have been generated by computer program (possibly a ML program).

If, in a reference interview, a reference librarian was given the keyword string 'black girls', they almost certainly would think that it is under-specified. A librarian would ask for clarification and disambiguation. Search engines sometimes do that, with overly long keyword strings, but generally they are in the realm of guesswork. Once a search engine has an initial return click among first list of links, it can usually improve the suggestions. So, usually searching is a process, not a one-shot question and answer.

Also worthy of mention are autocomplete (autosuggestion), and 'trends'. Google, to take an example, will autocomplete a partial search string on the Google Chrome Browser. Here is what it did for the author on 10/27/2022 for 'black girls':

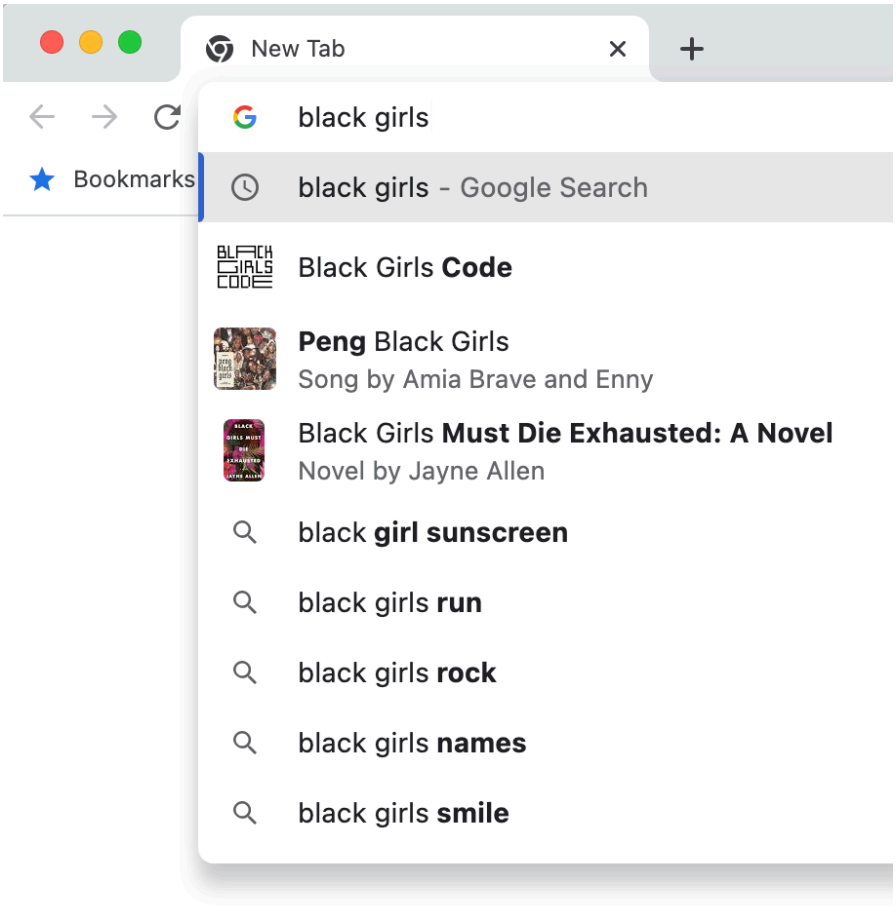


Figure 24. Screenshot of Autocomplete on the String 'black girls'.

We do not see anything untoward in that list. Google explains how autocomplete (and Trends) work on the page:

> https://support.google.com/websearch/answer/7368877?hl=en#zippy=%2Cwhere-autocomplete-predictions-come-from

In part, this reads:

> Autocomplete is a feature within Google Search that makes it faster to complete searches that you start to type. Our automated systems generate predictions that help people save time by

allowing them to quickly complete the search they already intended to do.
Where autocomplete predictions come from

- Autocomplete predictions reflect real searches that have been done on Google. To determine what predictions to show, our systems look for common queries that match what someone starts to enter into the search box but also consider:
- The language of the query
- The location a query is coming from
- Trending interest in a query
- Your past searches

These factors allow autocomplete to show the most helpful predictions that are unique to a particular location or time, such as for breaking news events.
In addition to full search predictions, Autocomplete may also predict individual words and phrases that are based on both real searches as well as word patterns found across the web.

Difference between autocomplete & Google Trends
Autocomplete is a time-saving but complex feature. It doesn't simply display the most common queries on a given topic. That's why it differs from and shouldn't be compared against Google Trends.
Google Trends is a tool for journalists and anyone else who wants to research the popularity of searches and search topics over time.

Notice here that autocomplete adapts to the User.

One general point to be made in all this is that the search engines are in competition one with another. They have incentives. In broad sweep terms, they need to be providing what the Users want, or need, or what is useful or valuable to them. Nowadays, Google has dominance with search engines. But that was not always so, and it does not have to be so going forward. There are various kinds of anonymous, non-tracking engines, and also straight-out competitors. In the absence of diktat, the Users will choose.

8.5 Social Media, Dis-, Mis- and False-Information

It is a jungle out there, and some ML programs have the potential to make the situation worse. ChatGPT can write English better than most English native speakers and writers. Essentially, readers would have difficulties in judging that ChatGPT English output had been written by a machine. ChatGPT could write disinformation tirelessly 24 hours a day 7 days a week. There is not much that librarians can do about this, apart from providing good education on information literacy. It may be that other ML programs, or even ChatGPT itself, could detect that samples of written English had been written by machine. This might be somewhat similar to plagiarism detection software. But being written by a machine does not of itself have to be bad. ChatGPT can abstract or summarize text. Summarizing today's newspapers, or this month's research journals, might be welcome and valuable.

8.6 Bias in the Organization of Information

8.6.1 Introduction

Traditional librarianship has devised such techniques as the ‘organization of information’ (content and container classification, abstracting, indexing, the use of surrogates, controlled vocabularies, thesauri, and the like) (Chan 2007; Rowley 2000; A. G. Taylor 2004).

There are questions of bias associated with these processes. To mention some:

- access to information, for individuals or groups, can be encouraged or discouraged
- straight out knowledge, or viewpoints, or theories, or beliefs or nexus of values can be conveyed or imparted
- attitudes towards the information resources themselves (including cognitive attitudes in the sense of social epistemology (Fallis 2006)) can be manipulated
- Mill's diversity of views (Mill 1869) can be promoted or obstructed
- Aristotle's diversity in the components of a good life (Wilburn 1999) can be promoted or obstructed

Potentially there are many topics that could be discussed here. We will restrict ourselves to a few.

[Appendix A has explanations of some slightly more technical librarianship terms that are used in Section 8.6:

1. emotive content: A.5
2. controlled vocabularies: A.2
3. classification and act of classification: A.6
4. taxonomies: A.1
5. thesaurus: A.1

]

8.6.2 Be Careful, and Sparing, with Emotive Content

The use of emotive content is valuable in advertising— it can help to sell things. It is also valuable in literature itself— it can manipulate our attitudes in liking or disliking characters and into becoming engaged with the story.

Librarians need to be careful with this, though. Any keywords, tags, subject labels, index terms, and so forth need to be low on emotive content. They need to be neutral. The reasons are: emotive content can produce harms of representation, and the use of emotive terms in indexes, say, or other stepping stones to content, will distort access.

ML systems should have little or no problem with avoiding emotive content. Large language models can just be prompted not to use emotive content. Some of the modern systems may be able to be given one example ('one shot') and then they will minimize emotive content.

8.6.3 Warrant and Controlled Vocabularies

A question is: where do the terms in a Controlled Vocabulary (CV) come from and what is the evidence or justification for introducing and using the particular terms that are used in a specific CV? There is a word or concept for this consideration. It is 'warrant' (Barité 2018). Warrant is important to us. It is one point where bias might arise.

There are different theories of warrant. We will briefly mention three: literary warrant, user warrant, and cultural warrant. Before explaining those, let us introduce a sample potential term for inclusion in a CV or some CVs : 'Gypsy Moth'. This is an example of a term that might have bias. Sabrina Imbler tells us in a 2021 article entitled *This Moth's Name Is a Slur. Scientists Won't Use It Anymore* that:

> The Entomological Society of America will no longer refer to common species of insects as "gypsy moths" and "gypsy ants," because their names are derogatory to the Romani people (Imbler 2021).

The article continues in part:

> For Ethel Brooks, a Romani scholar, the move is long overdue.
> As a child in New Hampshire, Dr. Brooks loved watching worms and caterpillars crawl across her hand. But one particular caterpillar, the hairy larvae of the species Lymantria dispar, terrified her. The larvae would swarm and strip the leaves from a tree, leaving behind so much destruction that people sometimes called them a "plague." But no one blamed L. dispar. Instead they blamed "gypsy moth caterpillars," the species' common name.
> "That's how they see us," Dr. Brooks remembered thinking as a child. "We eat things and destroy things around us."
> Dr. Brooks, now chair of the department of women's, gender and sexuality studies at Rutgers University in New Jersey, has spoken out against the use of the pejorative in fashion and college parades, she said. But Dr. Brooks never imagined the pejorative could be stricken from its use in the more staid realm of science.
> "It's hideous and super racist and it's hurtful," she said. "But what can you do about it?" (Imbler 2021).
> [To put some editorial interpretation here. Dr. Brooks is probably talking of the word 'Gypsy' as being 'super racist', not the phrase 'Gypsy Moth'.]

The name 'Gypsy Moth', or 'Gipsy Moth' (its British spelling) has not just been used for insects. Around 1930 the de Havilland aircraft company produced the Moth series of aircraft, which included the Gipsy Moth (and the related Tiger Moth, Puss Moth, Hawk Moth, Swallow Moth, Hornet Moth, etc.). When the supply of war-surplus eight-cylinder engines ran out, Geoffrey de Havilland designed the powerful and reliable four-cylinder Gipsy engine which was then used in the Gipsy Moth. The Prince of Wales owned a Gipsy Moth (you can see him portrayed flying it on the television series *The Crown*). Amy Johnson flew a Gipsy Moth single handed from Britain to Australia (roughly 11,000 miles at a flying speed of 100 mph).

> No other planes in their time and place so thoroughly served the advance of aviation (and civilization) in so many diverse ways as the de Havilland Moths (Harris 2002).

Separately, Gipsy Moth IV is the ketch that Sir Francis Chichester sailed single handedly around the world in 1967 (Chichester had worked earlier on the Gipsy Moth aircraft). It is reasonable to assume that Geoff de Havilland and Francis Chichester were not intending to demean their creations in any way by their choices of words 'Gipsy' or 'Gipsy Moth'. Quite the opposite, the names were intended to be positive descriptors. After all, de Havilland was selling mass produced airplanes. Then the items described, the engine, and the aircraft, turned out to be superlative. Additionally, Amy Johnson, the aviatrix, ' … was one of the most influential and inspirational women of the twentieth century (Gillies 2020)'. All was good for 'Gipsy Moth' in these domains, these periods, and these cultures. It is hard to imagine that any Romani were offended by 'Gipsy Moth' in the 1930s.

The idea of literary warrant comes from E. Wyndham Hulme at the beginning of the twentieth century. Lois Mai Chan et.al. summarize it

> ...the basis for classification is to be found in the actual published literature ... (Barité 2018; Chan, Richmond, and Svenonius 1985)

There is a slight difficulty or ambiguity here. What is 'actual published literature'? In 1911, that would have been published physical books that could be placed on shelves in libraries. But nowadays we have digital publication, e-Books, web pages, the Internet, and so forth. We need to take a wider view. Nevertheless, literary warrant would provide a motivation to using 'Gypsy Moth' in the 1930s, in the 2020s, and in the 2020s about historical 1930s literature, both in the sense of identifying an insect and identifying a training aircraft. Computers would be a great help here. The software does not necessarily have to be ML software. The processing has to look through the 'actual published literature' which it will be able to do really well (for all literature available in digital form). Notice here that with literary warrant there is one CV for all, whatever the area of focus is for that CV. The 'actual published literature' is the same for everybody. It is not one thing for one group of users, or culture, and another for a different group of users or culture.

User warrant focusses on the User. It is the User or the Patron (or, god forbid, the 'customer') that is trying to find the resources. Identifying the Users is not the easiest, nor is identifying how they do their searches. In the case of a public library, using say Dewey Decimal Classification, one might take the view that the Users are the 'public'. But it is also possible to be

more granular than this, patrons looking for materials on insects likely will be a different group to those looking for resources on historical aircraft. If the CV is for a smaller and more limited collection (e.g. for the index for a catalog for an aircraft museum), the tasks may be easier. Once again, computers are a great help. If patrons type into some kind of search box or Online Public Access Catalog or Discovery System, it will be easy to know what searches are being carried out. Nowadays most searches will be free text searches. The patrons can type in whatever they like and the software will learn what they do like. This time there can be several different CVs for the same collection of literature— one for one group of users of that literature and another for a different group of users of the same literature. Search software can personalize searches (as it would do for a private computer at home as opposed to in a public library).

Cultural warrant focusses on the cultures of groups of potential users of the resources. It is similar to user warrant in that there can be several different CVs for the same collection of literature— one for one culture of users of that literature and another for a different culture of users of the same literature. There can be obvious benefit in 'localizing' some CVs to place, time, beliefs, lifestyle, etc. i.e. to culture (no matter what literary warrant or user warrant might suggest in these cases). For example, legal systems can vary from state to state and country to country.

What are library practices here, and how can ML help (or hinder)? Typically, librarians will use universal systems, such as LCC, LCSH, DDC, to catalog, or provide metadata, for their resources. These CVs depend largely on literary warrant, sometimes with cultural adjustment. There is

widespread dissatisfaction with these CVs, particularly with cultural aspects. For example, Elizabeth Lorang et. al. write:

> Previous and ongoing collecting and description practices … were and are colonialist, racist, hetero- and gender- normative, and supremacist in other structural and systemic ways (Lorang et al. 2020).

Assume so. Were ML to use these practices going forward presumably it would just entrench them. (Mind you, if existing cataloging practices continue as is, without ML, there also presumably would be entrenchment.). ML does have the capabilities of correcting whatever shortcomings a CV might have. It certainly can reduce emotive content and provide cultural adjustment where required. Really some guidance is needed from librarians. For cataloging, librarians need to work themselves towards unbiased universal CVs (in so far as that is possible). ML can help with this (and also with the automation of insertion of values into the metadata fields of the resources).

What about 'Gypsy' and 'Gypsy Moth'? LCSH has this:

> Gypsy
>
> USE subject headings beginning with or qualified by the word Romani for topics related to the Romani people, e.g. Art, Romani; Romani poetry
>
> Gypsy moth [the insect, use as is]
>
> Gypsy Moth (Training plane)
>
> USE Moth (Training plane)
>
> (Library of Congress 2022)

This basically is literary warrant with some cultural correction. Notice, though, that using 'Moth' for 'Gypsy Moth' is asking for trouble with the precision of searches (because there are many other kinds of Moth training planes e.g. Tiger Moth etc.) Let us spell this out. You are interested in books on Gypsy Moth (Training plane), so you look up that subject in LCSH and learn that the catalogers have used Moth (Training Plane) for this topic, so you now search for Moth (Training Plane) and your search returns books on Tiger Moth, Puss Moth, Hawk Moth, Swallow Moth, Hornet Moth, Gypsy Moth etc. (most of which you do not wish to have). So, trying to be culturally sensitive, which we all want to be, has come, in this case, at the cost of ruining a search, and providing good searches is one thing librarians aspire to.

8.6.4 The Act of Classification Has Consequences

Obviously. In a court of law, classifying the accused as 'guilty' is rather different to classifying him as 'not guilty' (in terms of what the future might hold for him). So too for poisons, grades of scholarship applicants, and most everything else in daily life. (And, indeed, so too for being classified ‘black’ in (now fortunately historical) apartheid South Africa.)

Whole books have been written largely on this (cf. (Bowker and Star 2000)). Independently of the particular classification of items, we have noted how the accidental or intentional manipulation of the emotive content of equivalent concepts can affect attitudes to those things named.

Here are some of Sanford Berman's examples from the 1970s, and earlier, Library of Congress Subject Headings (S. Berman 1971):

> 'Yellow peril', 'Negroes, etc.', 'Mammies', 'Idiocy', 'Idiot asylums', 'Lunacy', 'Indians of North American, Civilization of,' 'Slaughtering and slaughter-houses—Jews', 'Barbarian invasions of Rome', 'Delinquent women'

Classification is also fallible, in common with all human cognitive endeavors. We know this directly from our everyday knowledge that anyone can make a mistake. We know it also from the test-retest unreliability of professional classifiers (catalogers) using precision and highly designed classification schemes (Snow 2017). If a classifier can classify the same item twice in two different mutually incompatible ways, classification is fallible.

The act of classification is also subject to bad-faith or 'rogue' cataloging. Everyone does this, all the time, pretty well every day. Faculty do it when they call a less than stellar student paper an 'A'. And some professionals do it too, fortunately not so often, perhaps when they have a private attitude, or strong feelings, which misdirect their work (and which, for example, might lead them to classify an information resource on abortion in such a way that the resource is difficult to find).

Classification is also subject to cultural factors. For example, different cultural groups have different attitudes to suicide, and thus different propensities to classify deaths as suicides, and this means that classification will not be isomorphic across cultures (independently of fallibility and rogue behavior).

In sum, the act of classification has consequences, can be used to produce attitude manipulation, is fallible, can be malfeasant, and has a dependency on culture. Some of the harms here are harms of representation, some are of allocation. Allocation can be seen as a harm of commission (doing allocation but not doing it fairly) or a harm of omission (failing to do allocation for certain groups or cultures).

8.6.5 Taxonomies Have Consequences

Taxonomies that have subclasses, i.e. most of them, make true or false assertions which are claims to knowledge. Asserting that one class is a subclass of another is a factual or conceptual piece of knowledge. For example, if a biological taxonomic scheme has whales as being a subclass of mammals, then it is offering the assertion that whales are mammals. Some of these schemes are intended to be objective i.e. they represent scientific or mathematical knowledge. Others are inter-subjective— for example, the Nurse Intervention Classification mentioned in Appendix A— and those represent decisions or conventions.

Some schemes can be problematic. Consider subjects, or topics, like 'creationism' and 'evolution' which we might be wanting to put into a hierarchical scheme of topics which allows them to inherit from some of 'scientific theory', 'false scientific theory', 'pseudo-scientific theory', 'religious view', and 'blasphemy'. What are we to put where? Well, who knows? Certainly, different groups of people would choose differently, and

different groups of people would disagree over which is a right or acceptable or appropriate classification. So, now any classification scheme does not so much contain knowledge, or, at least, uncontroversial knowledge, it does, however, contain or assert, a point of view (for example, one scheme might assert that creationism is a false scientific theory). There is a difference here between objective schemes and inter-subjective schemes. A mathematical classification that places the integers as a subclass of the reals is objectively right or wrong about that (and mathematicians can provide insight as to which that is). Similar considerations apply to the elements in Chemistry, or species in Biology.

Sanford Berman enlightens us that the 1970 Library of Congress Subject Headings are 'biased', and he means by this largely that they have unwelcome emotive content (S. Berman 1971). Berman also locates the bias. It lies with the classification schemes which are:

> …parochial, jingoistic Europeans and North Americans, white-hued, … Christian … heavily imbued with the transcendent, incomparable glory of Western civilization (S. Berman 1971)

Hope Olson echoes Berman telling us that the Headings reflect:

> …the exclusionary cultural supremacy of the mainstream patriarchal, Euro-settler culture' (Olson 2000).

And Bowker and Star wax long about generalizations of this: that the powerful subjugate the weak by imposing their will through classification (Bowker and Star 2000). (See also, (S. Berman 2000; Knowlton 2005; Olson 2002).)

Alright, but all schemes are 'biased'— biased in that they reflect a point of view (or knowledge, or beliefs, or opinions). We have to work with this. It is not an insurmountable problem. As Poincaré once remarked (about the indispensability of a point of view, when observing):

> He is no longer a slave who can choose his master (Poincaré 1905)

Then there are the conjectured conspiracies, surface or hidden, whereby the strong use classification to batter and imperialize the state of nature blissful. No comment on that here.

ML did not cause the problems in older library taxonomies (since they pre-date ML). However, care is needed to ensure that it does not prolong the issues. It may be used to counter-act them.

8.6.6 The Current State of Libraries and Their Organizational Systems

Melissa Adler writes:

> … libraries are complicit in privileging and circulating ignorance—inhibiting rather than opening up bodies of literature as sources of various knowledges (Adler 2017, 2).
> [Editorial note: 'Knowledges', plural, is an unusual form which perhaps has some currency in the sociology, or cultural aspects, of knowledge.]

Elizabeth Lorang and fellow authors write:

> Previous and ongoing collecting and description practices, for example, were and are colonialist, racist, hetero- and gender-normative, and supremacist in other structural and systemic ways. These understandings are the foundation on which training and validation data will be created and assembled; they will become reinscribed as statements of truth, even as we elsewhere champion the potential of computational approaches to uncover hidden histories, identities, and perspectives in collections. To engage machine learning in cultural heritage must mean confronting these histories, committing to the hard work of acknowledgment and rectification, and not simply reproducing them and giving them a whole new scale of power. There should not be a future for machine learning in digital libraries that is not first and foremost committed to, in the words of Thomas Padilla, "responsible operations" and to all of the ongoing, cross-cutting work that responsible operations entail. (Lorang et al. 2020)

The Thomas Padilla work alluded to is (Padilla 2019) *Responsible Operations: Data Science, Machine Learning, and AI in Libraries* and in turn this cites influence from Rumman Chowdhury (Chowdhury 2023). Padilla writes:

> Chowdhury defines responsible operations as collective investments in, " . . . processes to combat algorithmic bias." (Padilla 2019 Note 6) [Seemingly, the original sources no longer exist.]

Padilla's Report is a research position paper for OCLC. It is substantial. We will just briefly mention the section on Managing Bias. Padilla writes:

> Bias management activities have precedent and are manifest in collection development, collection description, instruction, research support, and more. Of course, this is not an ahistorical frame of thinking. After all, many areas of library practice find themselves working to address harms their practices have posed,

> and continue to pose, for marginalized communities. As libraries seek to improve bias-management activities, progress will be continually limited by lack of diversity in staffing; monoculture cannot effectively manage bias. Diversity is not an option, it is an imperative (Padilla 2019, 9).

Then he suggests the holding of meetings, holding symposia, convening working groups, etc.

8.6.7 Designing Information Taxonomies for Librarianship

Classification and classification schemes are important in helping Users meet resources. Professional librarians, and similar, are *users* of classification, not *designers* of classification. Almost no librarians produce classification schemes, and, with the advent of copy cataloging, truly few librarians classify resources.

Here are some of the balls to be juggled. Classification is the most effective if it is in terms of the concepts, world-view, and values of the Users ('User warrant' as opposed to 'Author warrant'). But rarely are the Users a homogenous group. We all of us are simultaneously members of many different groups (male/female, old/young, Republican/Democrat, filmgoers, those interested in sports, gay/straight, etc.). And many of these groups truly define 'a culture'. We are all simultaneously members of many different cultures. Which culture, or cultures, is the classification designer aiming at? Presumably, something in the middle. A starting point is for the designer to wind down any emotive content in the scheme, and wind up the correct cognitive content. A quiet toned neutral, always fallible, cultural

absolutism would be the starting point. Less emotion and more knowledge generally facilitates access.

But designing is not so easy. Let us talk websites for a moment. Consider trying to design a website for the Flat Earth Society. Now, the classification, in terms of that culture, would reflect a point of view, indeed a somewhat extreme and false point of view, not knowledge. So, there is a switch here to cultural relativism. Folk not part of that culture, normal folk, might struggle with using the resulting website. But the site might not have been designed for them, and perhaps this does not matter. Normally, the duties of stewardship require that the Information Professional get the data right (for example, with Credit Card Bureaus). But a classification designer might have the need to get the data as the Users believe it to be, not as it is. The designer might also be asked to increase emotion or manipulate attitude, and not to be neutral. What would be wrong with the Luxury Jeweler's website classifying some of their diamond rings as 'Truly gorgeous', 'Prince Charming's choice', 'Sugar daddy's specials', etc.? But then there are cases like a White Supremacists' Web site. Here, let us guess, what the Users want is something expressing false views in an extreme fashion, possibly even hate speech.

The absolute limits seem to be freedom of speech. If what is envisaged is protected speech, ethically or legally protected speech, then the designer can instantiate it. He or she may have personal misgivings, in which case they should just not undertake a task of the kind that gives rise to worry. There can also be professional and societal misgivings that add caution to freedom of speech, as we will see. Of course, there are many ways

computers and websites can change attitudes, for example, there is ‘captology’— the persuasive use of computers (Fogg 2003).

8.7 Navigation: Metadata Supported and Otherwise

Conceptually, metadata for an individual information resource seems to be just a table of field-value pairs, where semantically the field-value pairs cover container metadata (such as Print Date, Physical Location (if these apply)) content metadata (such as Subject Matter) and mixed container-content metadata (such as Author, Title). All this seems to be entirely independent of navigation by a User from information resource to information resource, if such an operation is appropriate.

However, this is not quite right. Quite a lot turns on the nature of the values used in the field-value pairs. Obviously, one resource may have the same author as another resource and a patron may want to navigate to other books by the same author. Then dates, as values, have obvious relations to each other (e.g. earlier date, later date, same date). A similar point is true of locations. Then for metadata values for fields like subject matter, the values would likely come from a controlled vocabulary, possibly a thesaurus. Typical thesauri will support generic, instance, and partitive relations. In turn, these support navigation following these relations. For example, a book might have a metadata subject field value of 'rhopalocera' and a patron might want to navigate to another book, or books, on a similar but more general topic— of course, the topic is 'lepidoptera' and that topic will be a metadata field value on other books. The metadata is here supporting navigation, but it is doing so in conjunction with a thesaurus or

classification scheme. If that classification scheme is 'biased', the actual navigation might be distorted.

In the traditional case, using information resources which were paper pages, structured and co-located into books, placed on shelves, placed in libraries, placed in buildings, the mere physicality of the information resources enhanced certain styles of information resource to information resource navigation (for example, the patron browsed along a shelf from one structured information resource to the next). The use of surrogates (such as cards in card catalogs) liberated this to a degree (you could have several cards for each book and thus, using the indirection of the surrogate, you could make a book have as many near neighbors as you wished). When computers are used as a tool in classification and navigation, generalization is complete. A computer can easily transform views and presentations and it can easily provide different views of the same computer 'information resource locations'. Computers also can and will make free and extensive use of surrogates. For example, web links, or hyperlinks, are surrogates, and there can be thousands of links to the same one web page; and, in turn, those links can be organized and displayed in various ways. The designer can make it easy or difficult to navigate from information resource, or collection of information resources, to another information resource, or another collection of information resources. Navigation, browsing and reading navigation, is a filter.

Navigation and search pose similar problems. They can be used to manipulate access. The designer needs to be competent enough to know what might happen. Generally, from diversity considerations, one wants to

widen access. But there are many problematic cases discussed in librarian literature:- access to dangerous true material (dangerous to the User (e.g. 'how to commit suicide'), dangerous to employer (e.g. 'how to hack a web site and steal credit card numbers from it'), dangerous to society (e.g. 'reservoirs near you and how to poison millions of people using them'), access to inflammatory false material (eg Holocaust denial literature), and many other cases besides (Froelich 2004; Hauptman 1988; 2002; M. M. Smith 1997; Wolkoff 1996). Traditionally, librarians have been fairly 'hands off'. They argued that they have no assured knowledge as to the uses that an information resource will be put to, therefore, as policy, they can just supply information resources and not worry further. Nowadays, they are more cautious. ML should be able to help, no matter what the policy is.

8.8 Ethical Arguments to Underpin Assertions of Harms of Bias

There are ethical concerns that are the province of all informational professionals, such as: freedom of speech, freedom of access, privacy, intellectual property, stewardship, and the like. There are Codes of Ethics for various professional bodies associated with librarianship. Such codes usually build off distinctions between personal and professional ethics, duties to an employer, to the profession, and to society as a whole (American Library Association 2021; American Association of Law Libraries 2019; Society of American Archivists 2020; IFLA 2012). These can be a great help. More fundamental than any code would be the principles of non-maleficence ('do no harm'), autonomy ('let people choose for themselves'), informed consent ('give people the requisite information

for choice'), and perhaps even the Golden Rule ('treat others as you would like them to treat you'). And more fundamental still would be rights, duties, and ethical consequences.

8.9 Annotated Readings for Chapter 8

Bowker, Geoffrey C., and Susan Leigh Star. 2000. Sorting Things out: Classification and Its Consequences. Cambridge, MA: The MIT Press. (Bowker and Star 2000) Berman writes about this: "... the work is crippled by its own density and almost occult, inaccessible language. It is dizzyingly awash in definitions and theoretical formulations, too often stated in impenetrable infosci jargon (S. Berman 2000)." That said, the book has important material to offer on the power of naming, and the good and the bad that classification schemes can do. It has extensive examples from diseases, viruses, tuberculosis, race in Apartheid South Africa, and nursing.

Ziegler, S. L. "Digitization Selection Criteria as Anti-Racist Action." The Code4Lib Journal, no. 45 (2019). https://journal.code4lib.org/articles/14667. (Ziegler 2019). This has an extensive bibliography (with links to further bibliographies).

Chapter 9: What Might Natural Language Processing (NLP) Bring to Librarianship?

9.1 Introduction

Natural Language Processing (NLP) is a subfield of computer science and artificial intelligence. It is concerned with enabling computers to interact with natural languages such as English, French, or Chinese. It involves the use of algorithms and computational models to analyze and derive, and produce, meaning from human language in both written and spoken forms.

NLP made a huge step forward with the advent of Large Language Models (LLMs), such as:

1. Google's PaLM2 2023 Pathways Language Model 540 billion parameters (Narang and Chowdhery 2022). There is now AudioPaLM which is a large language model that can speak and listen (Rubenstein et al. 2023) There is PaLM-E for robots (Driess 2023).
2. Databricks's Dolly 2.0, 2023
3. Meta's LLaMA family (Large Language Model Meta AI), 2023
4. Microsoft's XLNet family (Generalized Autoregressive Pretraining for Natural Language Understanding), 2019
5. OpenAI's GPT-X family (Generative Pre-trained Transformer 3, 3.5, 4), 2020, 2023

Not all NLP relies on machine learning, or, indeed, on LLMs, but a good proportion does so.

While Natural Language Processing primarily addresses natural languages, quite often the resulting software can be applied to images, videos, or sounds. Such software is 'multi-modal'. This comes about because the underlying substrate is just numbers. Text is reduced to numbers, and so can images, for example. Then some of the algorithms are equally at ease with numbers sourced from text, numbers sourced from images, numbers sourced from sound recordings, numbers source from video, and so on.

NLP is becoming of increasing import to librarianship.

9.2 The Pre-Processing Pipeline

An NLP program typically will some pre-processing of the text prior to doing its actual task. Here are some of the steps that might be carried out:

- Format Normalization. This is the conversion of the text to a desired standard format (e.g. by making it all lowercase and removing redundant spaces or special characters).
- Word Tokenization. This will split the string or stream of characters into words or tokens. Usually this would involve looking for spaces, or periods, or separators between groups of characters.
- Base Form Normalization. Linguist often view words as consisting of a base form (a 'morpheme') plus, possibly, a prefix and/or a suffix.

For example, the word 'transportation' consists of (trans + port +ation). There are techniques to get the base form.

    - Stemming. This means 'reduced to their root form'. So, for example, 'consult', 'consults', 'consulting' would all be stemmed to the root form of 'consult'. The result of stemming does not always have to be a well-formed word; for example, 'change' and 'changing' would be stemmed to 'chang'.
    - Lemmatization. There is a similar technique to stemming— lemmatization— which does reduce tokens to a root word ('change' and 'changing' would be lemmatized to 'change').

- Parts Of Speech (POS) tagging. This identifies whether the individual tokens are nouns, or verbs, or adjectives, etc.
- There also can be the dropping or omitting of 'stopwords'. Words like 'a', 'an', 'the' etc. do not carry much information and so usually can be omitted.
- Dimension Reduction. More than a few times there are many more words or tokens in the source than are needed for the processing. For example, a novel might have more than 100,000 words in it, but an algorithm might need only 100 words, the right 100 words, to determine its genre. There are techniques for reducing the amount of input (and thus reducing the amount of time and the cost of the processing).
- In the case of LLMs, the tokens (the processing units that they work with) would often be larger than single characters but smaller than entire words.

These pre-processing techniques are also used occasionally within current librarianship. For example, in classical information retrieval one approach is to try to match a 'vector' of a query to 'vectors' of documents. With this, tokenization, stemming, dropping stopwords, and, possibly, a TF-IDF calculation, will have been done on the text of the documents to produce the 'vectors'. [To explain the TF-IDF calculation. A simple way of identifying which document, among many documents in a corpus, is relevant to a search is to consider how often a term appears. So, for example, a search for information about Toyota cars might look for how often the word 'Toyota' appears in the different documents. What is desirable is that 'Toyota' appears frequently in the document to be returned but not frequently, relatively speaking, in the other documents in the corpus. The calculation *Term Frequency Inverse Document Frequency (TF-IDF)* determines this.]

Going forward from the pre-processing, some of the later techniques will use a 'bag of words' approach. With this only the words (the tokens or lemmatized words) matter— what the words are and the number of times they occur. Other techniques will look at the grammatical structure of the text and perhaps try to parse some or all of it. Yet others might use text embeddings.

9.3 Text Embeddings and Similarity

We have already mentioned *word embeddings* in the context of Word2Vec— see Section 3.8 — and the problem there was to determine whether two different words had the same or related meanings. But there is

the similar, but more general, problem of determining the degree to which two entire text strings are related, or similar, to each other. For example, one of these text strings might be a search or query string that a User has entered and the other text string might be that of a complete document in a document collection— if these two are related we might theorize that perhaps the document has some relevance to the search. *Text embeddings* are one technique to address text string similarity. There are many different ways of producing text embeddings, and there are many different free and commercial software applications to do it (for example, Sent2Vec, FastText, Doc2Vec, or Gensim). Almost all of them will produce a vector (i.e. a list) of numbers to represent the strings and also provide a measure that reveals how similar two vectors are. So, as an example, OpenAI's text-embedding-ada-002 model embeds the string 'Stochastic Psittacosis' to the vector:

> [0.012791729532182217,-0.009504193440079689, -0.007625600788742304, -0.012044000439345837,-0.012828806415200233, 0.012532186694443226, -0.005901498254388571,0.003066616365686059, -0.00211805081926286 2, -0.0020809732377529144,-0.002658763900399208, 0.024434056133031845, 0.002084063133224845, -0.02558345906436443 ... where there are about 1500 further numbers in the list.]

The source texts strings can be of almost any length. Although it would be usual to split up long texts into chunks. So, for example, a text document could be split into chunks (into chapters, pages, paragraphs, sentences etc.) then vector for a search string could be matched into vectors for those chunks and the relevant chunks have a location or locations within the document. There are some cautions. Modern embeddings can be produced by LLMs. Some LLMs have been trained on data produced or written before

2020. This means that such LLMs might not perform embedding well on more recent text (text involving recent events, slang, or changed practices of speaking or writing). Then, once you are in the world of LLMs there might be bias. So, care is needed is needed to check LLM embeddings for bias. (There are tests to do this.)

Once there are vectors for strings, and a similarity measure, several further opportunities become available. The context here is a corpus of text documents which has been embedded. Additional infrastructure would include a vector store, which is a database of the embedded vectors.

9.3.1 Searching by Meaning (Semantic Search)

We are all familiar with 'Find', the string-in-string search tool which is close to universal in word processing, or text editing, software. What this does is to find occurrences of the target word, a keyword, say 'attorney' for example, in a document. But if 'attorney' is the search string it will not find 'lawyer' (i.e. a synonym). But, in contrast, embeddings can search by meaning. They search by similarity of vectors. The vector for 'attorney' (or for phrases in which it appears) will be similar the vectors for phrases with 'lawyer' in them. This is a very powerful addition to search. What it amounts to, roughly speaking, is search with thesaurus support built in. This, for example, allows for search by topic or subject matter. Also, similarity admits of degrees, so the results returned can be ranked by how similar they are to the query string. This is a type of ranking by relevance.

Semantic search can improve on keyword search even in cases that do not directly involve synonyms. Here is an example from (Fitch 2023). Consider a user interested in the topic 'the fall of John Major'. [John Major was a prominent British Prime Minister in the 1990s]. Were the patron to ask a librarian about this, the librarian would understand exactly what was being sought and likely would be able to find suitable material. But a keyword search simply would not work because, for example, there would be many different ways of saying 'the fall of' and, separately, many document sources with the words 'john' and 'major' in them. Searching by meaning should or would do better and that would be carried out by checking similarity of an embedding of 'the fall of John Major' to embeddings within the documents in the relevant document collection. There is a striking second example provided by Amr Kayid and Nils Reimers of Cohere (Kayid and Reimers 2022). They report that Elastisearch (a keyword search engine) returned to the query 'what is the capital of the United States?' an article on Capital Punishment. (This happened because of occurrences of the (key)words 'capital' and 'states'.) Semantic search would not make this kind of mistake.

9.3.2 Research Trails

Once there is matching of chunks to chunks, then research problems or topics can be followed automatically from chunk to chunk, document to document, far and wide. This also might be useful outside of pure academic research. As examples, it might help with legal cases or with settling patent priority disputes.

9.3.3 Classification

Embedding can support classification in different ways. It can facilitate clustering. For example, the task of dividing a document corpus into 20, say, clusters of similar documents can be carried out by dividing the accompanying vector database into 20 clusters of similar vectors. It can facilitate classification using a supplied classification schema such as a thesaurus, controlled vocabulary, or ontological vocabulary. It could do this by seeing which vectors in the vector database of the texts were similar to the embedding vectors of the schema vocabulary.

9.3.4 One Style of Recommendation

If some sample 'desirable' documents are supplied, from a User, a group of Users, or from an institution such as a library, then embedding provides an easy way of finding other documents (i.e. recommendations) similar to the provided ones and of ranking the recommendations.

9.3.5 Plagiarism Detection

The embedded vectors can reveal if two vectors are too similar (maybe even that they are identical). This perhaps would suggest that the one source is plagiarized from another.

9.4 Named Entity Recognition

Named Entity Recognition (NER) is the ability of the software or agent to recognize in a source text the references to particular entities, e.g. to London or to Sherlock Holmes or to the Korean War. NER can also usually classify the entities into people, places, institutions, etc. The references need not be in some canonical form. For example, one journal article might have the word 'London' in it and another article the phrase 'the capital of England'— these are two references to the same entity. NER using a large language model, such as BERT or a GPT-X, on news articles, research papers, social media posts, etc., is extremely accurate. As you would expect, it is not quite as good on new, or extremely rare, entities. However, it is plenty good enough for most applications and it is distinctly better than humans attempting the same task. There is a difference here between the NLP model essentially recognizing or creating the relevant ontology and ontological vocabulary for itself against being provided with an ontology and controlled vocabulary to work with. For example, chemists have an ontology of elements, and compounds, and NER could use that when analyzing research publications in chemistry. Accuracy is higher if the NER is provided with an ontology to use.

NER is valuable for librarians, as it allows a User to obtain, for example, all the articles on London in a collection.

9.5 Topic Modeling

Topic modeling can look at a text or collections of texts and determine their 'topics'. Then it can 'tag' those texts with their topics. There is a slightly older way of doing this, and a slightly newer way which would use LLMs and embeddings.

With the older style, topic modeling is done by unsupervised learning. So, the ML software is clustering the contents of the texts. In the general case it needs be told how many clusters to make, and it does not have any names for those clusters. It needs to be guided as to those also. The result is that various different parts of the same text, or various different texts, can appear in the same cluster, meaning these parts or texts are on the same topic as each other. This type of analysis could also be done on documents, on paragraphs, on sentences, or on parts of sentences. To explain this at the level of documents. There are some background assumptions here, and some common techniques. The assumptions are a) that each document is a mixture of topics b) that each topic is expressed using a certain vocabulary, i.e. using certain words, and c) that any word has a certain probability of belonging to the vocabulary for a specific topic. Then the calculation is merely a matter of counting the frequency of particular words (say the number of tokens of the word 'car') and seeing how the best clusters of words can be formed doing justice to the word clusters and the documents. Common techniques include Latent Semantic Analysis (LSA), Latent Semantic Indexing (LSI) and Latent Dirichlet Allocation (LDA). (The word 'latent' here means 'hidden' or 'in the background'. See, for example, (Manning, Raghavan, and Schütze 2009) for an explanation of these

techniques. LDA can classify each document as a mixture of topics (Blei, Ng, and Jordan 2003).) This type of analysis is very quick. So, the analyst can certainly experiment by varying the number of topics to find a result that is acceptable.

Were modern style embeddings to be used, the general approach would be similar but the algorithms would use embedded vectors from a vector store.

The main use-cases of Topic Modeling are outside librarianship. As an example, security agencies may have an interest in which new threats are appearing in social media posts. Librarians, in their professional work, would typically have existing classification schemes that they would prefer to use.

9.6 Text Classification Problems

As we have explained under Section 9.3, certain kinds of text classification problems can be addressed using embeddings. But there are other techniques. There are a number of text classification problems that share a common ML approach. All that is needed are suitably labeled training data, produced in conjunction with the desired classification scheme, and then the ML system can classify the actual target data.

9.6.1 Shelving and Subject Classification

Text classification is similar to topic modeling, but it uses a supplied classification scheme. This means that the training of the ML system will be different. It will use supervised learning, and that means that there has to be labeled training data (which is always difficult to obtain). Modern systems are reasonably accurate.

Classification, especially shelving classification, is something different to subject classification, indexing, or CV tagging. With books, and Dewey or LC, for example, a book can only have one slot (this is because it can have only one position on a shelf). This requires that the classification is exclusive and exhaustive and that each book carries only one, perhaps complex, 'tag' (i.e. its call number).

9.6.2 Sentiment Analysis

Sentiment analysis, at its most basic, can pick out the emotive tone of some text. This might be useful in a commercial setting, for example, to process reviews of a product, dividing them into favorable, unfavorable, and indifferent. It typically works using only 'bag of words', i.e. the word tokens that appear in a document. It ignores what those words are saying. There are training data sets available for sentiment analysis. In that setting, supervised learning would be used to train an ML sentiment analyzer. You can see how a bag of words approach can go wrong with a naïve analyzer.

Say some example favorable words are 'good', 'brilliant', and 'excellent' and a text passage is written using negations only e.g. 'the product is neither good, nor brilliant, and definitely not excellent'. The review is unfavorable, but naïve sentiment analyzer might rate it as favorable because it contains only favorable words.

Modern large language models, such as BERT and GPT-X are very good at sentiment analysis. They would not use a bag of words approach. Also, they would not be trained using supervised learning. Likely they would be trained using self-supervision followed by reinforcement learning. Sentiment analysis might help librarians in any situation where they are trying to obtain feedback or opinions on items or policies or services or courses of action. The relevant patrons or constituents could be invited to provide their input, which then could be processed in whole or in part automatically using sentiment analysis.

9.6.3 Author or Genre Recognition

Most of the LLMs can work in 'one shot' or 'few shots' mode. The means that their user or programmer just needs to provide a few examples and then the LLM will be able to carry out the task that is illustrated. In turn, this means that merely by providing some examples of an author's work (or of books of a genre) and the LLM will be able to recognize which books are written by author (or of which books are of which genre).

9.7 Controlled Vocabularies, Thesauri, and Ontological Vocabularies

[Appendix A provides some background on the concepts used in this section.]

ML/NLP can create controlled vocabularies, thesauri, ontologies, and ontological vocabularies for any corpus of text. It would be able to do so swiftly. Semi-algorithmic techniques for producing these kinds of vocabularies are well known (see, for example, (Zeng 2005)). In recent years, computer support would certainly have been used for these tasks. But natural language processing adds power tools.

As a sketch of some of the techniques that might be involved. The terms in the documents can be clustered. This would be similar to topic modeling. These clusters can be arranged in hierarchies, and favored labels attached to each cluster. That would produce a rudimentary thesaurus and controlled vocabulary Possibly, Latent Semantic Analysis (LSA), Latent Semantic Indexing (LSI) and Latent Dirichlet Allocation (LDA) will be used. This will reveal the important concepts that are in use (and their relations). There could be named entity recognition (picking out, for example, that 'London' is the name of a thing or entity). There likely would be parts-of-speech (POS) tagging— identifying nouns, verbs, noun phrases, etc. The results would be a thesauri or ontologies better than those produced without ML techniques (and produce quickly).

Controlled vocabularies, and similar, are still important even with free text search and search by meaning. They can provide an interface, or commonality, between different documents or collections of documents.

9.8 Indexing and Automatic Indexing

A simple example of an index that we are all familiar with is that of an ordinary back-of-the-book index. The index itself consists of a list of *entries*. The entries are composed of *headings* and *locators*. For our purposes, we can assume the headings to constitute a (controlled vocabulary) thesaurus— so there might be sub-headings and sub-sub-headings, etc. in a hierarchical arrangement. Each entry is a pair of a thesaurus term and one or more locators. In the case of a book, a locator would just be a page number. So, the index itself has an easily navigable structure— the top-level terms are in alphabetical or filing order. But the book, in so far as its individual fine-grained topics are concerned, is a jumbled disordered mess. Were a User to be interested in finding out where the ship Pequod is discussed, described, or alluded to, in the novel Moby Dick, a proper index would make this task trivial whereas doing it by looking through the book would be a considerable challenge. An index provides simple, readily understandable, access to points of interest in an information mess.

As to creating an index in the first place, Bella Haas Weinberg writes (mainly under the assumption that the indexer is a human being):

> An indexer must be something of a prophet— envisioning the concepts likely to be sought by users of a document, expressing those concepts in terms likely to be sought by users, and providing cross-references from synonyms and alternative spellings as well

> as links to related terms to assist users in finding all the information that is relevant to their topics of interest (Weinberg, 2009).

Notice here that Weinberg refers to indexing in terms of concepts. It is possible to do *derived* indexing, which uses only terms that appear in the document or documents, or to do *assigned* indexing, which uses the concepts i.e it is indexing by meaning. Assigned indexing is superior (' ...to assist users in finding all the information that is relevant to their topics of interest '). Assigned indexing addresses the problem of synonyms and homographs, and also those very difficult cases where a concept is alluded to but not referred to explicitly. Derived indexing is pretty well trivial from a computing point of view (it is string in string searching). Assigned indexing is another matter entirely as it seems to require 'understanding' the material.

> 'Good indexing permits good retrieval'.

As the quote reveals, Weinberg sees the main intellectual challenge of indexing as being that of creating the controlled vocabulary thesaurus of headings that is to be employed. The secondary problem of scanning the text to catch all the locations for the locators in the entries is relatively routine and easy.

Indexes can be surprisingly sophisticated. Indexes typically work on structured sources or across collections of sources. So, for example, an index for a book might tell you that material relating to the concept 'machine' appears of page 37 and page 39. This is to conceive of the book as

being a structured source, with pages which are themselves sources, and the index is finding the component sources. Or, as another example, an index across a digital collection of full text sources might inform you that a particular topic is to be found within a few specific sources. In sum, indexes enable access to works within works: articles within periodicals; short stories, poems, or essays within a larger work; or individual papers from a conference.

Cumulative indexes would use the same thesaurus to index across several different sources; for example, across all the monthly issues for a year's publication of a journal.

AI/NLP can improve traditional indexing. It can do standard indexing faster than human indexers and the result can be of higher quality. It can produce a headings thesaurus without difficulty. Then fleshing out the entries with the locators is essentially trivial. Many of the actual cases of indexing will use an antecedently provided thesaurus. For example, indexing medical journals would do this. The complete task then comes under the heading of *automatic indexing*. Another advantage to AI/NLP indexing is that it can provide locators to outside sources, for example to other books or texts with references to Pequod (the whaling ship in Moby Dick). Such pathways to wider resources are valuable for researchers.

9.9 Abstracts, Extracts, Key Phrases, Keywords, and Summaries

Abstracts usually either give the content in a shortened, sometimes structured, form, or point to or indicate what treasures are to be found in the work or works but without actually providing those nuggets. Single sources can be abstracted, so too can multiple sources (as might be the case with abstracting the news from several different news sources into one abstract or abstracting many reviews of a source to produce an overall viewpoint).

Prior to the 1950s abstracting was always done by humans. But with the research of Luhn and others attention was given to the possibilities of automatic abstraction (Luhn 1958). This work is a sub-area of Computational Linguistics. The techniques were certainly computational, and some may be classified as AI, but, until very recently there was not really ML involved. Automation is important because there are many more documents that would benefit from being abstracted than there are human abstracters to do the work.

Abstracting would generally be either abstractive summaries or extractive summaries. The former tries to understand what is in the source and then paraphrasing it in a shortened form. The latter extracts and condenses what is there, without necessarily understanding it, using statistical features and 'signposts'. To illustrate the typical statistical features that might be used. Material at the beginning and end of the entire text is important, material at the beginnings and ends of paragraphs is important, words (other than

stopwords (i.e. ‘the’, ‘a’, etc.) that appear frequently are important and so on.

Researchers in this area do have techniques for assessing their results. In some form or other, these are normally comparisons against abstracts that have been produced by humans. (See, for example, ‘Rouge’ i.e. *Recall-Oriented Understudy for Gisting Evaluation* (Lin 2004).) It seems fair to say, as of 2021, the computational abstracts are *not* of higher quality than human authored ones. However, the software approach is thousands, millions, or billions of times faster than humans. So, for example, it can produce abstracts real time e.g. it can abstract news sources as fast as those sources are conveying news. There is Machine Learning in much of recent research on abstracting, but usually the ML is not used on its own but rather is augmented by tried and true methods (Widyassari et al. 2020). There does not seem to be any technical problem as to why ML abstracting would just not get better and better. Likely NLP abstracting will be the equal of, or superior to, human abstracting sometime in the early 2020s.

Producing Key Phrases or Keywords is a similar, but simpler, problem to abstracting. If it is desired that the ‘Keys’ appear verbatim in the text, then challenge is like extractive summary. Often, though, it would be preferred that Keywords come from a Controlled Vocabulary (CV). If so, maybe none of a text’s Keywords actually appear in the text. For example, the non-CV word ‘car’ might be in the text but the preferred CV Keyword term ‘automobile’ is not. ML is a possibility both with extractive Keywords and with pure abstractive Keywords (e.g. Controlled Vocabulary Keywords). Extractive Keywords can be approached with unsupervised learning

(Mishra 2021). This is a well-worked research area, not because the researchers are trying to help journal editors with lead-in Keywords for articles. Rather, the computational linguists want to extract important words for further processing. Abstractive Keywords (e.g. Controlled Vocabulary Keywords) could be set up as a supervised Classification problem. The CV would be the classes or categories, and the classification would note perhaps the best 8 terms that apply. These lead-ins specifically would help journal editors, and researchers and librarians looking for documents.

A summary of content would usually be longer and more detailed than an abstract of the same content. Summaries often rephrase the content. Otherwise, abstracts and summaries would be fairly similar. Summaries, or abstracts, may have specific features. They may have Named Entity Recognition. If the source is fiction, a summary may contain a plot outline or an identification of the plot type or story type. Summarizing across multiple sources is also a form of synthesizing or aggregating those sources— summarizing across ten different articles on linear algebra amounts to one way of aggregating those articles.

How might these techniques help librarianship? Summarizing documents may directly help Users (which may be human, or software) to find relevant content. Summaries may improve information access tools, such as catalogs and databases, by becoming part of the metadata that is used to characterize information resources. One of the first books written by machine learning— *Lithium-ion Batteries*— is a collective summary of 200 research articles (Writer 2019).

9.10 Text Mining and Question Answering

Text mining can extract potentially useful information from text such as information regarding named entities, topics, classification features, genres, summaries, and sentiments of reviews— techniques that have been described in this chapter. Any or all of these types of information may help a User narrow an information search and thus make it more precise.

Text mining is often used in conjunction with other techniques for example, with sentiment analysis, or with recommendation systems.

Later we will look at Undiscovered Public Knowledge (UPK)— that is knowledge that is in books or libraries that is 'undiscovered'. Text mining helps discover it.

9.11 Machine Translation

This involves automatically translating text from one language to another. The value of this to librarianship is obvious. These topics and techniques are discussed elsewhere in this book.

9.12 Evidence

It has been asserted in this chapter that NLP is capable of doing this-and-that. Researchers in this field do have evidence. There are benchmarks that

the systems are tested on (NLP-progress 2022). In fact, there is usually competition between the systems on how well they can do on the benchmarks. There are questions over the accuracy of the results of generative systems. There are difficulties with the prompts: with ‘noise’, context, and ambiguity. Also, an LLM will not usually predict the same thing twice— thanks to the probabilities. So, it not clear just what an accurate result is where an LLM is given the task of writing a paragraph in the style of Jane Austen. [There is further discussion in Section 4.8.1 on hallucinations.]

9.13 This Is Not Magic

Several of the topics and techniques mentioned in this chapter might sound a bit esoteric. But, actually, the means for a programmer (or library technical services department) to produce suitable software is simple and has been readily available since 2023. The best approach would be to use Large Language Models (LLMs), and to use one that has a public Application Programming Interface (API). For example, Open AI's GPT-4 does so. Then a programming environment like LangChain can be used to 'program' or configure LLMs from their APIs. The resulting program to perform one of the tasks mentioned in this chapter might have about 50 lines of code in it. A professional programmer might write 7 lines of code a day (when, as a pre-requisite, they have to think about and research the problem). So, the first program might take a week to produce. But the second and subsequent ones could be written in about an hour each. Go to it!

Since November 2023, there are have been even faster and easier ways of doing some of these projects. OpenAI has produce the building framework 'GPTs' which is a simple no-coding-required system for personalizing the multimodal GPT-4 Turbo to various NLP and LMM tasks (OpenAI 2023d). Neither programmers nor LangChain are needed for this.

9.14 Text Processing and Laws

Needless to say, there are laws and contracts that might restrict the unfettered use of NLP on document collections such as libraries. Further, librarians and their institutions can be intimidated at the mere prospect of legal cases (like the rest of us). The laws concern primarily copyright and intellectual property, and they differ from country to country. This area is complex. A good initial source is H. Andrés Izquierdo's *20 Artificial Intelligence and Text and Data Mining: Future Rules for Libraries?* (Izquierdo 2022).

At a rough handwaving level, we might say this. Copyright concerns the *expression of ideas*, not *the ideas themselves*. So, when Einstein wrote the theory of relativity, his actual words might have had some copyright protections but the theory of relativity itself did not. So, when NLP abstracts, or paraphrases, or summarizes, or text-data-mines, documents or collections *in its own words*, it might be that there would be no copyright concerns. In contrast, extractive abstracts or summaries, or quotation of passages verbatim, etc. might be problematic. Additionally, many copyright laws have exceptions for 'fair use' which might include use for research, teaching, and non-commercial uses.

Contracts are another matter. The owners of the intellectual property can seek whatever contracts they wish. Librarians, or their institutions, can agree, or not agree, to these contracts, as they wish. Some advice: librarians should agree only to contracts that are permissive on text-data-mining and other NLP techniques.

A further problem or issue is that more than a few times the owners of the intellectual property are unknown or untraceable (as might be the case with some historical documents).

9.15 Annotated Readings for Chapter 9

Izquierdo, H. Andrés. "20 Artificial Intelligence and Text and Data Mining: Future Rules for Libraries?" In *Navigating Copyright for Libraries*, edited by Jessica Coates, Victoria Owen, and Susan Reilly, 497–540. De Gruyter Saur, 2022. https://doi.org/10.1515/9783110732009-022. (Izquierdo 2022). Just scan this.

Jurafsky, Dan, and James H. Martin. "Speech and Language Processing," 2023. https://web.stanford.edu/~jurafsky/slp3/. (Dan Jurafsky and Martin 2023) This is a standard text. It is probably too advanced for us. The draft of the 3rd edition is available free on the web.

NLP-progress. "Tracking Progress in Natural Language Processing." NLP-progress, 2022. http://nlpprogress.com/. (NLP-progress 2022) This describes the 'state-of-the-art' in the subfields of natural language processing.

Chapter 10: What are the Opportunities for Librarians?

10.1 Introduction

In 1989, Edward Feigenbaum ('the father of expert systems') observed in the paper *Toward the library of the future* that the problem with the then extant libraries was that the books did not talk to each other (Feigenbaum 1989). He continued:

> ... imagine the library as an active intelligent knowledge server. It stores knowledge of the disciplines in complex knowledge structures, perhaps in a knowledge representation formalism yet to be discovered or invented. It can reason with this knowledge to satisfy the needs of its users. These needs are expressed naturally with fluid discourse. The system can, of course, retrieve and exhibit. That is, it can act as an electronic textbook, but it can also collect relevant information, it can summarize, it can pursue relationships. It acts as a consultant on specific problems, offering advice on particular solutions, justifying those solutions with citations, or with a fabric of general reasoning. If the user can suggest a solution or an hypothesis, it can check it. It can even suggest extensions, or it can criticise the user's viewpoint with a detailed rationale of its agreement or disagreement. It pursues relational paths of associations, to suggest to the user previously unseen connections. Collaborating with the user, it uses its processes of association and analogizing to brainstorm for remote or novel concepts. With more autonomy, but with some guidance from the user, it uses criteria of 'interestingness' to discover new concepts, new methods, new theories, new measurements (Feigenbaum 1989, 122).

Feigenbaum is addressing a certain kind of library here, what would be called an 'academic library' or a 'research library' (and those categories might include university libraries and medical libraries).

There are many types of libraries, including:

Academic libraries
Children's libraries
Digital libraries
Medical libraries
National libraries
Public lending libraries
Reference libraries
Research libraries
Special libraries
University libraries

[See (Wikipedia 2023i; American Library Association 2007) for a description of some of these and an explanation of their functions.] The librarianship activities associated with these are many and varied, and there are also other librarianship activities not connected with institutions specifically of these types. For our purposes, as a practicality, we have to restrict our gaze. For the most part, we will be looking at machine learning in connection with scholarship, research, and advancing knowledge (i.e. with Feigenbaum's approach of relating AI to the notion of a library as a knowledge server). This means that our main focus will be academic libraries (including university libraries), medical libraries, and research libraries. We also often consider librarianship in general. There may be the odd remark on machine learning in other kinds of libraries (as examples, that humanoid robots may be valuable for children's story times in public libraries (Nguyen 2020), and that handwriting recognition may be valuable

for the Vatican Archive, which is a special library (D. Firmani et al. 2018)), but coverage of these areas is going to be thin. Sorry.

There is the idea of collections as data and data as collections, with librarianship as an interface (see, for example, (Padilla et al. 2019)). Standard libraries can be thought of as collections— collections of texts, documents, and books, and also, perhaps, of means of access to the same or similar items. Some, or many, of these collections, or parts of collections, will be born digital, or become digitized, and will be available to computers, artificial intelligence, and ML. Thus, there is the notion of collections as computer data. But also, we now live in the age of big data. Huge amounts of data are being accumulated by researchers, governments, social agencies, and commercial interests. Many subsets of this big data are collections. They are libraries. They are subject to the ordinary concerns of librarianship, such as: organization, preservation, storage, access, retrieval, and stewardship. So, there are collections as data and data as collections, with librarianship as an interface. For example, researchers in astrophysics, with their telescopes, radio dishes, and myriad of other instruments, produce data repositories that stand in need of librarianship. Then, still in their day jobs, these researchers read research papers in journals and collections. Provision of these also needs librarianship.

There is an abundance of modern digitized, or born-digital, resources, and this abundance is growing rapidly all the time. There is a plethora of sources spewing ever more 0s and 1s. Facing up to this on behalf of librarianship are a relatively small number of expert human librarians. There is an order of magnitude difference here— the potentially valuable

tasks, and the collection sizes, far outweigh the capabilities of a team of human librarians, even if the number of librarians were to be increased a million-fold. As an example, which is becoming slightly dated, there is the MeSH indexing of biomedical publications (about 7000 articles a day were being indexed). Yuqing Mao and Zhiyong Lu wrote:

> MeSH indexing is the task of assigning relevant MeSH terms based on a manual reading of scholarly publications by human indexers. The task is highly important for improving literature retrieval and many other scientific investigations in biomedical research. Unfortunately, given its manual nature, the process of MeSH indexing is both time-consuming (new articles are not immediately indexed until 2 or 3 months later) and costly (approximately ten dollars per article) (Mao and Lu 2017).

There is a general point to be made here (the same general point as will be made elsewhere over and over). Computers have a 24x7 work ethic, and ML can often supply expertise. Many areas have been automated already. But this is an ongoing and expanding process. Of course, human librarians use many tools to increase their capabilities viz-a-viz librarianship, but we are moving into an age where the ML systems on their own can produce excellent performance (maybe even superior performance).

Librarians already work with many systems that have connections to AI and ML. Here is one way to classify librarians working in AI or ML in roles:-

Librarians can be seen as

- Synergists
- Sentries

- Educators
- Managers
- Astronauts

Individual librarians may fulfil different roles on different occasions, or, indeed, be working in different roles at the same time.

10.2 Librarians as Synergists

Librarians have several thousand years of experience of working with recorded information. They are ideal partners to AI to bring out the best in this information with all its aspects, challenges, and facets. And, on the other side of this, there are many AI and ML technologies that have the potential to improve librarianship. There is an opportunity for synergy or symbiosis.

We will introduce a few sample possibilities here and expand on the topic later in a dedicated chapter.

Synergists:

- **Intellectual Freedom.** AI, ML, and librarianship have potential to enhance Intellectual Freedom, both the expression of free speech and access to it: as examples, Optical Character Recognition leading into machine reading and speaking of text, and, separately, machine translation of text leading to recorded text being available in many

languages (Knox 2023). Machine learning is now expanding these possibilities to hundreds, even thousands, of languages.

- **Smartphones.** Many young, and not so young, people use smartphones as their main means of access to information. This opens an obvious invitation to librarians to bring libraries and librarianship to patrons via smartphones (e.g. via a bundle of voice and sound, static and dynamic text, and images and video). This might involve AI, ML and chatbots.
- **Improving Intermediation Between Users and Information Resources.** As examples: search engines rank their returns, from more important to less important, and, separately, there are 'recommender' systems which can recommend other resources similar to ones favored by the User. Pure librarianship does not have either ranking or recommending in any developed form. Even the computer, or AI systems, versions have not yet realized their potential.
- **Improving Traditional Cataloging, Classification, and Retrieval Tools.** Standard point: there are, and always will be, fewer expert human catalogers than are required to address the increasing flood of resources. Computers and ML can redress this. Also, NLP can perform valuable librarianship tasks that are not practical for humans. For example, it could look at a million publications and identify, *de novo*, what subjects, topics, or genres each might be labeled with. 'De novo' here means 'without using any antecedent classification schemes'.
- **Chatbots.** Chatbots have the potential to do most of the current tasks where librarians interact with patrons, either synchronously or

asynchronously (e.g., in person, on the telephone, on video calling, by messaging, etc.).

- **Release, Produce, Curate, or Inspire the Production of, Training Data.** Librarians already have metadata on the contents of libraries— metadata which, for the most part, is accurate and labeled well. Also, librarians are well placed to use handwriting recognition on archives of historical documents. Developing handwriting recognition for these will require the production of data (most likely labeled samples).
- **Social Epistemology.** The promotion of social knowledge— social epistemology— is a vital function of librarianship (Egan and Shera 1952; Fallis 2002; 2006; Fuller 1988; Goldman 1999). It is a function that librarians have been doing for millennia. Social epistemology faces problems aplenty nowadays with disinformation, misinformation, fake news, deep fakes and the like (Meszaros and Goodsett 2022). Librarians, in conjunction with the tools of ML, are well placed to take on the challenges. There is now the opportunity, and the need, to do more.
- **Images.** There are libraries and collections of images (for example, the Center for Creative Photography in the University of Arizona (CCP 2020)). Also, many publications contain images, figures, and diagrams. Attaching metadata to the images, and then finding the desired images, has always proved difficult. But now ML is allowing it to become possible for standard librarianship operations to be performed on media like images. For example, you can now enter an image itself, or a verbal description of an image, as search 'terms', and a suitable search system will be able to find all similar or relevant

images within a document or collection. These possibilities would be enriched by insight from librarians. Images are getting to be addressed reasonably well by ML.

Jason Griffey asks an interesting question in connection with synergy. He introduces it by way of recommender systems and improving personal intermediation:

> … the system trains itself from the user's behavior. One can easily imagine systems built to do this sort of automation work for researchers and students. As AI systems continue to be easier to implement, having a system local to your device that learns your preferences, your interests, and your needs will be commonplace. Researchers and students will have AI systems that find sources for them, summarize them, help them build bibliographies, and more. Over time, these systems will become irreplaceable archives of the learning and thinking history of individuals, a sort of universal diary of their activities. Now, imagine for a moment that this sort of system exists and is used by most learners. *Who would you prefer be the developer of such a system: a large corporation like Facebook, or a collaborative effort by educational institutions and libraries?* (Griffey 2019, 27) [Italics added.]

We will see an example later of 'a collaborative effort by educational institutions and libraries'— that by Kent Fitch in the Section 10.2.4 (Fitch 2023). Another example should perhaps be mentioned, that of Kaushik Roy and fellow authors using retrieval-augmented generation (RAG) (Roy et al. 2023)

10.3 Librarians as Sentries

Unfortunately, many of the potential benefits of ML and librarianship have concomitant downsides. Here the librarians can be sentries. To anticipate, the challenge is that advances in ML have been so rapid that suitable ethical systems, laws, and policies either do not exist or are out of date. Librarians can help create these.

Sentries, here are some examples:

- **Copyright and intellectual property** Intellectual Freedom interacts with restrictions of privacy, intellectual property, state secrets, and so forth. These considerations required careful management (and librarians have plenty of experience with, for example, intellectual property, fair use, and licensing).
- **Bias management** There are various kinds of bias that can arise in connection with ML and information provision. Librarians do have experience in managing bias. For example, they do it in collection development, collection description, instruction, and research support (Padilla 2019).
- **Monitoring techniques to improve search** Methods associated with personalization and recommendation impinge on privacy (by, for example, monitoring Users' behavior to create the personalization). Filtering has had a bad reputation within librarianship (primarily due to misadventures involving filtering schools' access to websites). But, when doing a search, providing

recommendations is filtering. Then filtering can lead to information silos or bubbles. There are problems here to be addressed.

- **Intellectual freedom** This needs management. The collections and services should presumably give patrons access to a wide range of diverse and thought-provoking materials, while also protecting them from potentially harmful or offensive or false or ungrounded or obviously crazy content. But how to do this is a question: one person's crazy might be another's happy territory. And the idea of 'protecting' introduces paternalism which should not be required for fully functional adults. Machine learning systems could very easily work in a paternalist way.
- **Inadvertent censorship** The properties and behaviors of advanced machine learning systems for example, those built from foundation models, are usually not fully known. Caution is needed to ensure, for example, that there is not accidental censorship.

10.4 Librarians as Educators

Librarians have a role as educators.

Educators:

- **Information Literacy** Librarians have always been the standard bearers for information literacy. But Artificial Intelligence (AI) has changed what information literacy can be. There is ongoing development of new tools for interacting with information, for example, personalized search. AI or ML, as research disciplines or

commercial enterprises, devote little or no attention to information literacy itself (except in so far as the AI or ML can be a part of any educational course or teaching on any discipline or topic).

- **Data Literacy, Data Science Fluency, and AI Literacy** There are other forms of information related literacy that are becoming important (Ridley and Pawlick-Potts 2021a; Digital2030 2022; Druga et al. 2019a; Carlson and Johnston 2015; Padilla 2019). For example, research scientists are often required to have data management plans. They are producers of data, and they need to know how to manage it for the benefit of other researchers and the world at large. Librarians can help the researchers directly and also play a role in educating student researchers in the management of data. Another example is that AI and ML have expanded the realm of Automated Decision Making (ADM) (e.g. the making mortgage loans). An informed citizenry should be alert to the strengths and weakness of ADM.
- **More Intelligent Consumers of Information**. This includes both patrons and library staff.
- **Better Informed Citizens** Outside of actual information literacy, there are considerations of helping citizens understand ML and computational aspects of the world they live in (and, in the case of living in the USA, how some other countries are approaching it). As examples, there is the Canadian *Algorithm and Data Literacy Project* (Digital2030 2022), and there is the European *Generalized Data Protection Regulation* (Wolford 2018).

10.5 Librarians as Managers

At a perhaps a more day-to-day level, librarians run libraries, both physical and digital. Computer assisted automation is widely used and is of obvious benefit. Book acquisitions, cataloging, serials control, and circulation, information retrieval and dissemination, interlibrary loan, cooperative acquisition and cataloging have been automated in the library (Lakshmikant and Vishnu, 2008).

AI can improve the running of libraries. We are trying to steer clear of plain automation in this text. We will try to restrict ourselves to cases where the software uses or simulate artificial intelligence.

Managers:

- **Workflow and Improving Service** ML has the potential to enhance productivity and efficiency in libraries. Many the components here have been mentioned already: ML cataloging, personalization, recommender systems, better search, chatbots for customer service, predictive analysis for collection management, user behavior analysis to improve service, and digitizing special collections.
- **Optimize the Use of Space (and, Indeed, Other Resources)** ML is good at optimization problems.
- **Robots** To put books back on the shelves (!), to do story-telling, to meet and greet, and more.

- **Mimic Librarian Experts' Behaviors** To support decision making and management.

10.6 Librarians as Astronauts

Astronauts, well, who knows? But most of human knowledge is in libraries. ML will allow exploration here of a kind that has never been done before.

Astronauts:

- **Creating Knowledge**. There is deep text extraction and synthesis from materials already in libraries. More than a few university researchers conduct their research using only their initiative and the contents of libraries. ML will be able to do this (and render the faculty researcher redundant in this regard).
- **Drawing Out Knowledge**. There are many special collections that have not been digitized and transcribed (and, perhaps, for some of those that approach might not be acceptable). But processed collections— with indexes, for example— might provide access to treasures.
- **Moonshots ? Who Knows What They Might Be?**

10.7 Annotated Readings for Chapter 10

[Several of these publications are out of date, as are many sections of the present text.]

Asemi, Asefeh, Andrea Ko, and Mohsen Nowkarizi. “Intelligent Libraries: A Review on Expert Systems, Artificial Intelligence, and Robot.” Library Hi Tech 39, no. 2 (2020): 412–34. https://doi.org/10.1108/LHT-02-2020-0038. (Asemi, Ko, and Nowkarizi 2020) This is a reasonable literature review. (Some of their references seem incorrect as to their topics e.g. it identifies Amin and Razmi 2009 as being on the topic of 'knowledge-based indexing'.)

Bourg, Chris. “What Happens to Libraries and Librarians When Machines Can Read All the Books?” Feral Librarian (blog), 2017. https://chrisbourg.wordpress.com/2017/03/16/what-happens-to-libraries-and-librarians-when-machines-can-read-all-the-books/. (Bourg 2017)

Cordell, Ryan. “Machine Learning + Libraries.” LC Labs. Library of Congress, 2020. https://labs.loc.gov/static/labs/work/reports/Cordell-LOC-ML-report.pdf. (Cordell 2020). This has a good survey of topics, and a valuable bibliography (generously also provided as a Zotero shared library). There seems to be no mention or discussion in it of chatbots.

Cox, Andrew M., and Suvodeep Mazumdar. “Defining Artificial Intelligence for Librarians.” Journal of Librarianship and Information Science, 2022, 09610006221142029. https://doi.org/10.1177/09610006221142029. (Cox and Mazumdar 2022) This brings a different conceptualization to the interactions between AI and libraries to the one offered here.

Cox, Andrew M., Stephen Pinfield, and Sophie Rutter. “The Intelligent Library: Thought Leaders’ Views on the Likely Impact of Artificial Intelligence on Academic Libraries.” Library Hi Tech 37, no. 3 (2019): 418–35. https://doi.org/10.1108/LHT-08-2018-0105. (Cox, Pinfield, and Rutter 2019). There is a useful table in this of possible AI initiatives, relevant competencies, and 'alternative providers'. As to the latter— commercial interests, publishers, or university Information Technology departments may produce or provide the AI tools or services. Librarians watch out!

Dempsey, Lorcan. “Generative AI, Scholarly and Cultural Language Models, and the Return of Content.” LorcanDempsey.net, 2023. https://www.lorcandempsey.net/generative-ai-a-note-about-content/. (Dempsey 2023b)

Das, Rajesh Kumar, and Mohammad Sharif Ul Islam. “Application of Artificial Intelligence and Machine Learning in Libraries: A Systematic Review.”

ArXiv:2112.04573 [Cs], 2021. http://arxiv.org/abs/2112.04573. (R. K. Das and Islam 2021).

Fernandez, Peter. "'Through the Looking Glass: Envisioning New Library Technologies' How Artificial Intelligence Will Impact Libraries." Library Hi Tech News 33, no. 5 (2016): 5–8. https://doi.org/10.1108/LHTN-05-2016-0024. (Fernandez 2016). Library Hi Tech News has a column, occasional articles, written by Peter Fernandez. These are good, recommended.

IFLA. "IFLA Statement on Libraries and Artificial Intelligence," 2020. https://repository.ifla.org/handle/123456789/1646. (IFLA 2020) This is particularly good, recommended. It has useful references in its Annexures.

Jakeway, Eileen, Lauren Algee, Laurie Allen, Meghan Ferriter, Jaime Mears, Abigail Potter, and Kate Zwaard. "Machine Learning + Libraries Summit Event Summary," 2020. (Jakeway et al. 2020). There seems to be no mention or discussion of chatbots in this.

Padilla, Thomas. "Responsible Operations: Data Science, Machine Learning, and AI in Libraries." Dublin, OH: OCLC Research, 2019. (Padilla 2019). This has input from around a hundred knowledgeable practitioners and academics. It aims to develop '… a research agenda to help chart library community engagement with data science, machine learning, and artificial intelligence'. As such it is slightly different in aspiration to the present text (which tries to look at the intellectual challenges arising between ML and librarianship). It appears not to contain a mention or discussion of chatbots.

Padilla, Thomas, Laurie Allen, Hannah Frost, Sarah Potvin, Elizabeth Russey Roke, and Stewart Varner. "Always Already Computational: Collections as Data: Final Report," 2019. https://doi.org/10.5281/zenodo.3152935. (Padilla et al. 2019)

Rolan, Gregory, Glen Humphries, Lisa Jeffrey, Evanthia Samaras, Tatiana Antsoupova, and Katharine Stuart. "More Human than Human? Artificial Intelligence in the Archive." *Archives and Manuscripts* 47, no. 2 (2019): 179–203. https://doi.org/10.1080/01576895.2018.1502088. (Rolan et al. 2019) This is directed at archives and record keeping.

Chapter 11: Librarians as Synergists

11.1 Intellectual Freedom

Intellectual freedom is the right, and perhaps the ability, to access or disseminate any information that the person wishes. Now, there are many qualifications here, obviously, such as those concerning privacy, intellectual property, offensive materials, state secrets, etc. Let us build in the restrictions and consider only those cases where there is a right to freedom of access or to freedom of speech (i.e. to freedom of dissemination). Then let us concentrate on the *ability* to exercise our rights in those cases. It is reasonably common in rights discussions to distinguish *privilege* rights from *claim* rights (Wenar 2021). A person has a privilege right to do X, if, and only if, they do *not* have a duty *not* to do X. So, a mature person presumably has a privilege right to read, say, Huckleberry Finn, and that right exists because they do *not* have a duty *not* to read it. Additionally, a privilege right is a freedom, to the right's holder, which does not impose any obligations on any other party to facilitate that right. If you have a privilege right of freedom of access to some form of information, you may access that information, but no one is obliged to give you practical help to do so. Claim rights, in contrast, impose a duty on some person or entity to actively assist a person to exercise the right. For example, some folk might not be able to afford to buy, or simply might not wish to own, some of the books that they are interested to read— in these cases, libraries, perhaps public libraries, can facilitate a claim right of access to those resources. In contrast, book sellers or bookstores are not subject to a claim right of access

(in the absence of purchase, or similar, from the would-be reader)— they do not have a duty to provide access.

Libraries, especially public libraries, aim to satisfy claim rights of access. Some also may provide a physical public space for meetings. This would be assisting with disseminations, perhaps even assisting with claim rights concerning dissemination. Libraries, or librarians, also often are active in opposing 'banned books'. Third parties might want copies of, say, Huckleberry Finn, removed from a library. What is happening here is that the third parties are partially denying the claim rights of others (and possibly also denying the privilege rights of others). Librarians are defending these rights. So, librarians help meet some intellectual freedom rights and defend others (Garnar and Magi 2021; American Library Association 2008).

How machine ML help with this? The short answer is: librarians need to provide direction as to what they wish their policies to be by way of addressing rights of access and dissemination. In the past they have provided such direction, and they continue to do so in the present. But the landscape as to what is possible is changing rapidly, so policies require ongoing attention. ML cannot really help with what the policies should be. But once the policies are in place, ML can certainly help with the implementation. Here are some details.

11.1.1 Text Recognition

Text recognition has been an absolute triumph in enabling people with visual challenges, or reading difficulties, to access written and printed materials. In a library setting, text is understood to be recordings that are written, or printed, or reproduced, pieces of language that can convey meaning. These recordings can range from handwritten documents to professionally printed books, from documents thousands of years old to those from the present day, from originals to photocopies of photos of photos pasted into a painting collage. The recordings have a form, and mostly they have meaning. Text recognition is trying to capture the structured form of the text as computer input, and not the deeper meaning. For example, consider the text ‘Now, I am angry.’ What text recognition needs to be able to do is to pick the letters in sequence i.e. ‘N’, ‘o’, ‘w’, etc., to pick the words (likely from the separators such as the blank spaces), to pick the sentences (likely from the upper-case letters, periods, and punctuation), etc. It does not need to be able to grasp what the text means (e.g. what or who the indexicals like ‘I’ and ‘now’ refer to, or what the word 'angry' means).

Two further points should perhaps be made. Text recognition's main line of business is not that of helping people who are visually challenged, rather it is that of helping large corporations and institutions convert their paper records and data into digital form. Assisting the visually challenged is happy side effect of that enterprise. Second, complexities arise with OCR over whether the source documents are monolingual or multilingual, and

consequently the various alphabets, orthographies, and grammars that might be in use (see, for example, (Alpert-Abrams 2016)). It is possible for target sources to be multilingual. They would constitute additional challenges. Multilingual sources are not common. But they certainly can appear in older materials where there might be a mix of a scholarly language (e.g. Latin) with a vernacular (e.g. Italian, French, English.) Consider an older handwritten text, written in several languages, and which uses 'loan words' (i.e. uses or quotes words from other languages). Adequate OCR here might require relevant outside knowledge (for example, of the provenance, and cultural background, of the document).

As mentioned earlier, most modern publications exist at some point in digital form. If there is access to that digitization, text recognition is not needed. So, generally speaking, text recognition for library purposes is not as important now, 2023, for modern materials, as it was, say, for materials produced prior to the 1980s. Also, the use of a variety of fonts and printing styles, which make text recognition harder, are a modern development. That help is to some degree offset with other problems with older materials such as difficulties with ink fading, poor printing, deterioration of the paper, etc. Nevertheless, the main problem areas for text recognition in libraries are older printed works, where the publishers have a restricted choice of font palette, and, separately, handwritten documents. Generally, printers are trying to print works in a form that the readers can read. They are not trying to distort letters left and right, or larger and smaller in way that would challenger readers (or, indeed, OCR systems). The original OCR systems of the 1970s would not have been ML systems, but nowadays they

certainly would be. For a discussion of OCR, we can consider just neural nets.

Conceptually, text recognition takes place in two phases: the optical character recognition (OCR), and the discernment of structure within the stream of characters. From an ML point of view, OCR is an example of a supervised classification problem. First there is a classification system, which is a collection of classes, or categories, or sets— in this case, that is going to be an alphabet of characters e.g. ‘a’, ‘b’, ‘c’ etc. Then there will be the characters, or character instances, themselves which will start computational life as visual disturbances, marks, blobs, etc. on paper or some other medium. Likely they will then become patterns of pixels in electronic images that the algorithms can address. Processing will individuate these into characters-of-interest, then ML will classify each one of these as being an ‘a’ or a ‘z’ or a ‘j’ or a space or other character.

It sounds simple, but it is not. There are issues and challenges. Let us start with the classification system. There are about a dozen writing systems, and hundreds of alphabets (Ager 2023). There is the need to pick one or several here. Which one is chosen presumably depends on the purposes at hand. If a library has primarily English sources, presumably the English Alphabet would be a good start. This is not in any way to insult or denigrate, for example, Japanese and Hiragana. There are two points to be made here. Classification systems usually carry baggage with them. They have assumptions and consequences that extend further than the classification systems themselves. (See, for example, (Bowker and Star 2000; S. Berman 2000).) An OCR implementation might be able to recognize English

characters but not Hiragana— and that might matter for those who can or cannot benefit from the specific OCR. Second, the choice of classification system or systems might involve a wide range of constituents, not just programmers or AI researchers imposing their will. The patron, user, librarian, research challenge, and infrastructure and legal framework, also can provide input.

Then Supervised ML OCR is going to be taught, or learn, how to classify characters. It will be supplied with a *training set*, which will be a reasonable sample of letters and the right labels or classifications of what they are. The training set might run to 100,000 labeled characters. The overall technique is an optical one, so it is the features of the sample letters that can be detected optically that will be the input (e.g. size, shape, color, grid arrangement of component dots or pixels, etc.). Then the program will attempt to correlate combinations (i.e. vectors) of these with the correct classification e.g. that a particular sample token character is an 'a'. More than likely, the program will make many mistakes initially. But either the programmers, or the program itself, will tune various parameters (e.g. weights on the components of the vectors) to improve the classification until it reaches an acceptable level of performance. Typically here, the neural net would have 5-6 layers and hundreds to thousands of neurons in the layers.

The training set needs to be adequate for the task. For example, if the letter 'j' does not appear in the training set, it is unreasonable to expect the ML program to classify js correctly. Even if js appear, there needs to be enough of them in the various fonts and scripts (cursive or not, monospaced or

proportional, etc.) for the program to be able to learn what is correct and what is not. OCR, i.e. the task of recognizing the actual individual characters, would not usually be an end in itself. Rather, the interest would be in the words that those characters form, or, more generally, the text.

If the OCR, or Text Recognition, application has access to a wider context, that can improve its performance. For example, if the ML is recognizing entire words from their component characters, and separators, then the first letter of 'On' is going to be the letter upper case 'O' and not the numeric letter zero '0'— the number zero makes no sense in that context.

Let us assume going forward that there is ML that can take an input of images (i.e. a page of visual representations of characters, an ordinary book of words, etc.) and produce as output text. Now, text, in a computer science sense, consists of 'strings' or sequences of characters, and, once in that form, they can be processed in a variety of ways. For example, they can be searched, or edited (cut, copied, pasted, transformed etc.). To give a practical example of the advantage of strings over raw images, finding the word 'covid' in some (computer science) digital text is near trivial for a computer program. Contrast that with the following. Imagine a photocopy, or photograph, of the front page of a newspaper, and the problem finding whether there is a sub-image in it that might be construed as an image of the word 'covid' — i.e. does the word 'covid' appear on the front page of the newspaper? That is a much harder problem. (You know that, of course, from the online Captcha tests that are used to detect whether you are a human or a program or 'bot' pretending to be a human (Wikipedia 2023a).)

OCR for modern printed monolingual text is near perfect. There is a qualification here. OCR needs training data. There are about 7,000 languages in the world, and about half of these have writing systems. Of those 3000 or so with writing systems, quite a lot less have enough printed text to be suitable training data. Current OCR systems can read about 200 different languages. There will be more languages than that with suitable training texts, but there also needs to be either commercial or intellectual incentives for the relevant OCR research to be done.

As the Text Recognition systems have improved, their compass has been extended to include handwriting recognition i.e. transcribing handwriting to the 0s and 1s of computer text. This is very important. To give an example. The Vatican Apostolic Archives (the Vatican Secret Archives) contain hundreds of thousands of documents going back many centuries (D. Firmani et al. 2018; Wikipedia 2022f). Most of these documents are handwritten, and certainly more than a few of them are of great significance. As examples, one is from Henry VIII to the Pope requesting a marriage annulment, there is the Catholic Church's 1521 excommunication of Martin Luther, and there are notes from the trial of Galileo.

Transcribing handwriting is of a level of difficulty harder than transcribing printed text. There are different cases to be considered. There is personal handwriting to be transcribed 'online' (i.e. as it is being written, real-time, perhaps onto a smartphone or tablet). For example, a User may handwrite entry into a text-messaging app. There is purely personal, or official, handwriting to be transcribed 'offline' (i.e. from a recorded document after it has been written). Transcribing online is easier than offline because

information is available on the pen strokes, their sequence and timings. This information helps, for example, with segmentation: with identifying the lines, the words, and the characters. There can be real-time transcription with the characteristics of offline, for example, so-called scene-in-scene transcription. For example, driverless cars may have the need to 'read' road signs and other textual information that they are 'seeing'. There may also be some value in the ability of a driverless car to read handwriting within its video stream. Some restaurants display a menu outside with the day's fare handwritten on a small blackboard. Search applications might want to have the ability to read this from a video feed. For our purposes, with recorded documents, our interest is primarily with offline transcription.

> The problem of automatic Handwritten Text Recognition (HTR) persists since document digitization started. Text recognition is a simple task for humans, but it has been proved to be complex for automatic systems. In fact, it is considered an unsolved problem and under active research.... The high variability between writers and the cursive nature of the handwriting text are the main difficulties presented by this problem. These difficulties have meant that historically, the practical applications of offline handwriting recognition technologies have been quite limited (Sueiras 2021).

To a degree, everyone's personal handwriting is different. Nevertheless, mostly, people are trying to communicate with their handwriting, and they have learned, have been taught, or are required, to write in a way that their writing can be read. There are differences in the writers, their writing styles, and their purposes. There are differences in the intended roles of the product documents.

Some cases are easier to transcribe than others. There is cursive handwriting and block handwriting (writing separate letters). For example, international arrivals at an airport may be required to hand 'print' (i.e. block handwrite) their flight and passport details into a form, where some of the fields of the form are required to be text, others known to be dates, and yet others known to be numbers. Such handwritten forms can be machine read quite easily, even though there may be different authors and writing styles involved.

Documents like those in the Vatican are often official documents of one kind or another, written by scribes. In cases like these, not only does the scribe personally want the document to be easy to read and definite in form, but so does the scribe's employer (the government, the Queen, the seller of the land, etc.). Scribes need to get it right or lose their jobs (or maybe their heads). Scribes in a certain cultural setting are usually required to follow a specific style (roughly: they have to write in a certain font). Most of the medieval and later documents in the Vatican will have been written in the Carolinguan Miniscule font (or later fonts related to it).

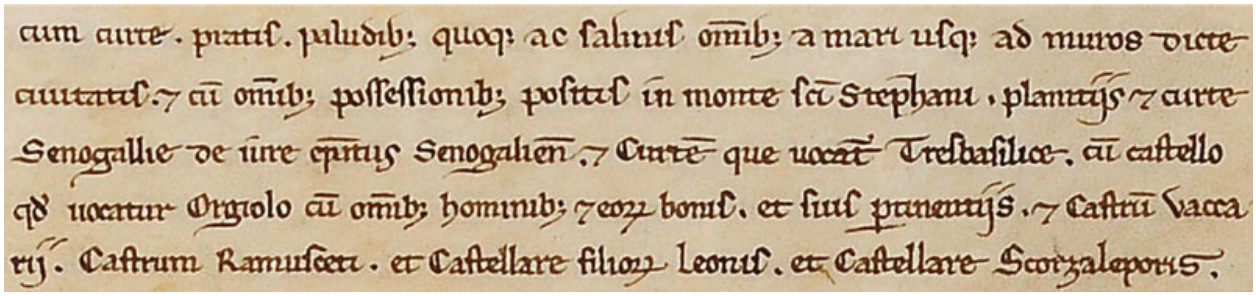


Figure 25. Sample Text From the Manuscript "Liber septimus regestorum domini Honorii pope III", in the Vatican Registers (Donatella Firmani, Merialdo, and Maiorino 2017).

Many cases are much more difficult than this. There is a manuscript collection of the works of the English philosopher Jeremy Bentham. (Bentham wrote on philosophy and law. He is known mainly for proposing Utilitarianism.) Bentham wrote the manuscripts largely himself. But he also had helpers who wrote some portions. He wrote mostly in English, but occasionally with pieces in Greek or French or other languages. He made corrections, but often leaving the originals and the crossings out in the text. He also sometimes wrote in columns, or with included passages.

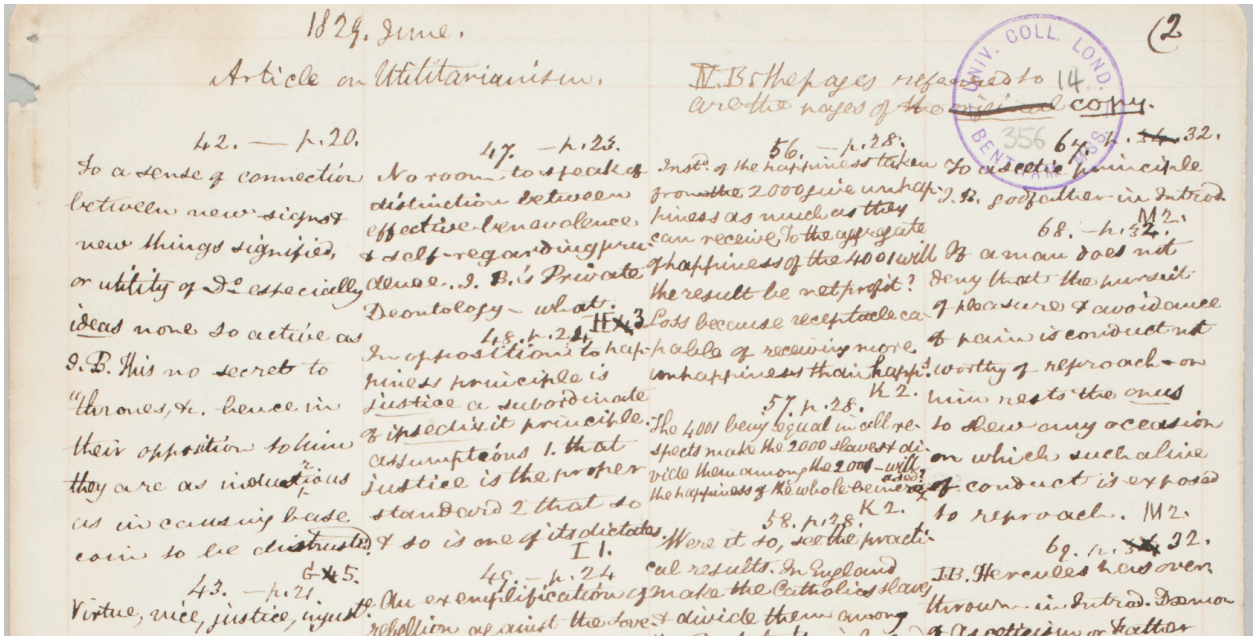


Figure 26. Sample From a Manuscript by Jeremy Bentham (University College London).

However, coming at this from a different direction, think back a few years of your own experiences. A regular High School teacher might receive thirty handwritten essays from students and be able to read them without any difficulty. (If there was difficulty, likely the student would be given remedial handwriting instruction.) This suggests that ML would allow computers to read cursive or block handwriting from a variety of authors. Typical approaches to this in 2023 would likely involve Neural Nets or Deep Learning, and this means, basically, that they could be covered by a Foundation Model.

11.1.2 Speech to Text

We are all familiar with voice-controlled assistants like Alexa, Siri, or Google Assistant. These can take speech, or dictation, and first turn that into (computer) text. Usually, the way that this step is done is that an ML program will attempt to recognize and classify the phonemes in the input sound. The phonemes are the smallest, or atomic, sound components of speech in a language. In English there are about 44 of these. In response to dictation, the program will produce a stream of phonemes. Converting this stream into well-formed and sensible text is hard. But it has been done now with 99% accuracy for most well-known common languages. To give some idea of the difficulties. In English, there can be the same letters producing different phonemes (‘thin’, ‘these’), different letters producing the same phonemes (‘sit’, ‘city’, ‘eight’, ‘ate’). To resolve these, context is required—i.e. what makes sense in the wider sample of speech. Google's Cloud Speech-to-Text V1 supports dictation in hundreds of languages (Google Cloud 2023). Apple’s software for iPhone covers dictation capabilities for about a hundred languages or their dialects (British, American and Australian English, are all listed separately, for example). Meta's Massively Multilingual Speech (MMS) project is aiming at speech recognition and text-to-speech models that can recognize over 4000 languages and work with over 1,100 languages (Meta 2023). Meta also open-sources its models (which may be good). Of course, recognizing a language is one thing, being able to take dictation in it at speed is something else.

The training here is largely with audio recordings which have transcripts (obviously here resources are meagre with rare or unusual languages). If the target language has a text-to-speech converter, that can be used to create data.

Speech-to-text can be used to transcribe speeches and lectures. Text-to-speech can be used to create audiobooks. The two technologies can be used to help people learn languages. The two technologies can be used to help people communicate in their preferred language. They can help preserve language diversity and prevent languages from 'dying'. Languages usually die from a lack of use and native speakers. This may come about for a variety of reasons (speakers favoring other languages, government policy, etc.). However, the technologies create an infrastructure which retains the knowledge of how a language is to be spoken, written, and read.

There are settings where either typing or writing cannot be used or in which they are not the best option. Dictation is usually taken to be much faster than typing or writing, maybe 3-10 times faster. [There is evidence counter to this in certain circumstances such as start to finish form-filling in a medical or legal context.] Some situations— for example, if there is background noise— are not the best for speech input. But the choice is there. Voice commands, voice input, and the ability to ask questions verbally increase intellectual freedom.

Smartphones are ubiquitous. They can take text input and provide visual text output (and some users are astonishing adept at typing their text

input). They also can work with sound and will usually have a voice-controlled assistant.

Transcribing spoken language into text makes it easier to search. This may be useful for libraries that have audio resources (such as historical recordings of Question-and-Answer sessions to a parliament, or oral histories.)

11.1.3 Sign Language to Text, and Text to Sign Language

There are AI programs to translate sign language to text. These will use some means of capturing or 'viewing' the images, perhaps a smart phone. Then they will classify what is seen. This may be done real time— that is, as the same speed as the signs are being given i.e. at conversation speed. But classification of, say, the hand movements is not the only challenge. There are maybe 300 sign languages, and they have different characteristics, of course. Usually, a sign language does not consist solely of signs made with the hands. Rather there is a wider range of gestures including facial expressions and bodily movements. This makes the translation problem much more difficult.

Within libraries, and other repositories of information, text, images, and other visual artifacts are the primary display medium. If a person is challenged visually, then text-to-speech technologies are very helpful. If a person is challenged aurally, but not visually, then sign-to-text technologies are certainly valuable. But the visual text itself would presumably be

available to such a person, and that would provide a means of access to recorded information. There will be cases where sign-to-text technologies would help in a library setting— for example, for those who can sign but not read or write. Libraries do have holdings in sound; for example, podcasts, recordings of lectures and speeches. Transcription to text might help there.

We are all familiar with important addresses and speeches being signed simultaneously with being spoken. Obviously, that is a valuable access and inclusion technique. But when the focus moves to recordings, say of the Australian Prime Minister, Julia Gillard, saying, in 2012, 'I will not be lectured about sexism and misogyny by this man...' , there does not seem to be the same need for a signed version. Recordings of the speech are available in video, sound, and text.

11.1.4 Helping Filter and Personalize

A problem highlighted by Jorge Luis Borges's short story *The Library of Babel* is that just having access to absolutely all information would likely prove to be useless because the relevant information would not be able to be found (Wikipedia 2023o). Intellectual freedom needs a filtering of garbage, and a personalization to provide relevance. ML can provide that. [This topic will be revisited later.]

11.1.5 Scholarly Publishing

This is a small aside, but it is an example. Researchers in the western world are more-or-less required to publish their research papers in English. Many are not native English speakers. There is an awkwardness at this point. Some scholarly publishers provide automatic translation to English for all submissions (see, for example, (CHOICE Media Channel 2022)).

11.1.6 What Can Be Done With Computer Text

Suppose, now, that we have the computer text corresponding to a page, or pages, or books of printed or written characters, or of speech, or of signs. What can be done with that? A lot, see Section 1.4.

11.1.7 ELI5 Translation

It is common to see the acronym 'ELI5' on discussion hosts like Reddit. ELI5 means 'explain it like I am a 5 year old'. An interesting and fascinating point is that LLMs can do this. They can explain passages as though the audience were 5-year-olds. But also they can simplify and re-render passages at any level whatsoever from 5-year-olds to the level that the passage was written at original (and even go in advance of that perhaps to a more sophisticated version).

This matters for two reasons. It increases intellectual freedom, because it makes difficult passages accessible to all, and it personalizes the delivery of information into a form that the user would like or to a learning style that the user needs.

11.2 Improving the Intermediation Between 'Users' and 'Information Resources'.

11.2.1 Some Users Might Not Be Human

Some Users might be ML programs, or software tools that are employed by the human User, or employed by other programs or 'bots'. To a degree, this is true already. For example, many repositories will publish their metadata for 'harvesting'. (See, for instance, the *Open Archives Initiative Protocol for Metadata Harvesting* (Wikipedia 2023k)). What is happening here is that services are being built off resources that a library or libraries have, or off services that libraries offer. Librarians can help this by ensuring that their holding are accessible in relevant ways to appropriate 'Users'. Accessibility in this context will likely involve considerations of licensing and intellectual property. One type of service that likely will become very important is that of text mining. We will look at text mining later.

11.2.2 Some Resources Might Not Be Resources

Some resources might not be resources in the sense of being physical or digital texts or documents. They might be services. A library might provide a service, for example a certain kind of access to its holdings. A second library might provide a similar or different services. A further party might bundle those services into another service.

11.2.3 Digital Archiving

Archiving is a little different to straightforward librarianship in that it deals almost exclusively with historical materials and its organizational principles are often different (for example, in giving pride of place to provenance). Digital archiving— preserving digital content for future use— is different yet again. It is one of the many things that commercial and governmental institutions do. For example, digital archiving may be mandated for compliance reasons— e.g. keep your tax records for 10 years!

11.2.4 Enhanced Search Engines

Existing search engines already use machine learning. Companies like Google typically do not reveal exactly how their systems work— presumably for commercial reasons. But some techniques are known or can be surmised. Google search uses RankBrain, Hummingbird, Panda, Penguin,

Pigeon, etc., along with the original PageRank algorithm (Wikipedia 2023l; 2023m; 2023f). Some, or maybe most, of these use ML. What these techniques concern mostly is with ranking the results that the search engines find. Apparently, there are 200 or so ranking factors that can be taken into account. Some of these are to do with the web pages themselves (whether they contain real content or are 'click bait'). Some concern the users and their search histories. Some even concern the locations of the search origins. For example, in the US the word 'boot' picks out a footwear item you wear, in Britain the word 'boot' can mean this but it can also mean what Americans would call the 'trunk' of a car. The search engines know this and can react accordingly. The search engines also use NLP more generally, and there is considerable amount of machine learning in NLP.

Commercial search offerings are close to being able to do the following in response to a query:

1. Semantic search, augmented by named entity recognition, to produce and return, in the first instance, say, ten links to relevant web pages
2. To summarize those pages at a length and intellectual level either suitable for, or requested by, the user
3. To load the contents of those pages into a LLM ‘database’, thus allowing the user to chat and ask questions about the web page contents, with the LLM answering (with citations)
4. To construct a knowledge graph from the contents, and from this produce topics or references of further interest. It could here produce a list of related searches.

The technologies in use here are:

- Semantic search (from NLP, embeddings, and LLMs)
- Summarization (from NLP and LLMs)
- Question answering (from NLP, probably vector stores of embeddings, and LLMs)
- Knowledge graphs (these are infrastructure in this setting— what they are is explained in Appendix E)

One example of a 'librarianship paper' showing work of this kind is Kent Fitch *Searching for Meaning Rather Than Keywords and Returning Answers Rather Than Links* (Fitch 2023). Fitch writes:

> Large language models (LLMs) have transformed the largest web search engines: for over ten years, public expectations of being able to search on meaning rather than just keywords have become increasingly realised. Expectations are now moving further: from a search query generating a list of "ten blue links" to producing an answer to a question, complete with citations.
> This article describes a proof-of-concept that applies the latest search technology to library collections by implementing a semantic search across a collection of 45,000 newspaper articles from the National Library of Australia's Trove repository, and using OpenAI's ChatGPT4 API to generate answers to questions on that collection that include source article citations. It also describes some techniques used to scale semantic search to a collection of 220 million articles (Fitch 2023).

Librarians have expertise in information retrieval. This could be used to create search engines that are more effective at understanding and fulfilling user queries. There already are image searches, with image input or image

output. There are location-based searches. Google maps tells you of restaurants near you, not those on the other side of the continent. There are possibilities and opportunities here.

11.2.5 Personalization and Recommendation

What an information provision system is trying to do in response to a perhaps not well articulated request from an individual User is to supply that User with all and only the relevant resources (i.e. 100% recall and 100% precision) from a well-defined, or not so well-defined, collection of items. [Recall is the percentage of relevant items in the collection that are returned. Precision is the percentage of returned items that are relevant. Roughly, recall is signal and precision is absence of noise.]

There are explanations and qualifications required here. Martin Frické (2013) writes:

> Relevance is usefulness to the User as judged by the User. Relevance is just a user-controlled honorific that connects [items] and utility (on a particular occasion of retrieval). The Patron or User is ultimately the sole arbiter of relevance.

Assume so. This means relevance is subjective in the sense of being User and occasion specific. In turn, this means that retrieval is subjective, and the best retrieval systems will allow for personalization for individual Users. Separate from this, the boundaries of a collection may be somewhat wooly as they may range from a single bookshelf out to a single library from there out to the entire Internet.

ML can personalize provision and retrieval by knowing about a patron and a patron's past behavior. Traditionally, libraries have often had advisory services. These usually would consist of one or more librarians who would know the local collection and would have experience of patrons, their information needs, and the resources from the local collection that would meet those needs. The assessment or feedback on their work might be limited. There might be some surveys, or similar, but not much more.

Computer and AI 'advisory services' would or could know about many users, many collections, and would have extensive data and feedback as to how it was performing (and improve itself accordingly).

11.2.6 Recommender Systems

We are familiar with recommender systems from our experiences with Amazon, for books or other items, and with Netflix and streaming sites for movies. Recommender systems can improve reading experiences.

There are various possibilities here as to how these might work, which we can describe in a library setting. Data is needed, and pretty much the more the better. This might include:

- demographics about the User (e.g. whether they are a child or a senior citizen)

- any information that they wish to share about their likes and dislikes and preferences (genres, subject matters)
- their reading or access history
- information about the books or available resources (perhaps including their genres, abstracts, ratings, or reviews).
- other relevant factors
- diversity in plots, characters, authors, genres

There is collaborative filtering which puts the user in the context of other similar users and recommends on that basis. There is content-based filtering which pays attention only to the items that the user likes, or seems to like, and works with their properties (ignoring information about other users). Most systems will use a hybrid of both approaches. One is that an anonymous user accesses or views or reads a single item and remains anonymous (but might have a continuing single session identity). There is not much data to work on here. But the item itself will have many properties, such as author, subject, genre, length etc. It also might tie into explicit written reviews or feedback from other patrons or even professional reviewers or critics. Some sorts of recommendations might be able to be made here. The continuing session identity might give some feedback as to whether any of the recommendations were followed up on, by the central user, during that session.

More usual would be the setting where every patron's accessing history, and every item's accessed history, is known and recorded as data. (This data can be kept private, and identities not revealed.) This accessing history will be 'implicit' data about the items accessed. There also will be 'explicit' data

about these items, such as their authors, genres, subject matters, reviews, and citations or references or links to them. At this point, either just data about past interests might be used or data about that supplemented by data about other users and their histories. Any ongoing behavior by a user can be used to update their profile.

Then, probably, either one of two approaches might be made. The first is to put the patron into a 'stereotype' (i.e. a class or group) consisting of other patrons similar in respect to the patron seeking recommendations. This would be done largely on the basis of the present reading and the access history. The other is not to bother with classes and just to let a ML system look for similarities in reading behavior across users. It takes the likes and dislikes of the User being helped then overlays those on the likes and dislikes of other individual Users to produce a match. The upshot can be a recommendation system that helps users with personalized, perhaps ranked, recommendations as to resources that would be interest, useful, and relevant.

Such systems can do more. They might be able to predict how new items, yet to be purchased items, will be received by groups of users— the preferences. And thus, in the case of libraries, they can help with collection development.

11.2.7 Understanding What the User is Asking For

Internally, behind the curtain so-to-speak, a traditional information retrieval system will likely use Boolean queries or queries in a database query language like SQL. But few Users are competent to do input their questions or requirements in this fashion (see, for example, (Frické 2021)). Some Natural Language Processing here could smooth the interface between User and such systems.

11.2.8 Text Mining

As described in Chapter 9, text mining can extract a variety of potentially useful information from text such as information regarding named entities, topics, classification features, genres, summaries, and sentiments of reviews. Any or all of these may help a User narrow an information search and thus make it more precise.

Going a little broader, text mining can look through (usually large) corpora for valuable information or patterns. The size of the task makes it hard, if not impossible, for humans to do. We have discussed facets of this elsewhere— for example, creating an encyclopedia by ML requires text mining. Question answering, summarization, tracking research ideas, etc. all require text mining. We will discuss the topic again in the context of Undiscovered Public Knowledge.

One red flag or alert is over the question of licenses or the legal position over the mining of texts. In so far as they can, libraries ought to ensure that they can mine their holdings. (Unfortunately, the reality might be that some other entity will do the mining and charge the libraries for doing so.)

11.2.9 Information Assistants (and 'GPTs')

Let us adopt a form of thinking here. Let us characterize information intermediation in terms of *tasks* and *control* (or *flow*). Tasks consist of searching, recommending, paraphrasing, translating, etc. Control (or flow) is how the tasks fit together. There is *sequential* flow, which is where tasks follow each other in a sequence. There is *conditional* flow, which is where there is a condition (call it 'if') and if the condition is satisfied (i.e. is true) flow goes down one branch (one further sequence of tasks) and if not the flow goes down another branch. Finally, there is *loop* flow, which is where a sequence of tasks repeats or loops either a given number of times or until a condition is satisfied.

Given this structure, informal information algorithms can be constructed. For example,

> Is there a day of the week that the Musée d'Orsay in Paris is closed in July?
> Is there a day of the week that the Louvre in Paris is closed in July?
> If they are both closed on the same day, say what day that is, otherwise say which museum is open when the other is closed and what the relevant days are.

> Look up recent research on twisted spin.
> Summarize the best papers, no more than 5 papers.
> Present the summary at the level understandable by a graduate student in physics.

We can conceive of these in terms of flow and tasks, and so can LLMs. LLMs can take this kind of input in English, spoken or written, and answer it.

[Editorial Note. The first edition of this book continues:

> "The answers, July 2023, may be a bit rough. But it will be only months before the answers will be very good.
> In sum. Shortly there will be information assistants that can combine information tools on the spot. Users will be able to mix and match tools. That might not matter to library patrons on all occasions of their uses of libraries. But the lives of researchers are going to be transformed."]

On November 6th 2023, OpenAI announced 'GPTs' and the upcoming GPTs Store which will sell GPTs or provide them free. As mentioned earlier in Section 2.9, there is a builder technology that allows the construction of GPTs. GPTs themselves are relatively small assistants or agents based on the underlying LMM GPT-4 Turbo technology (or its successors). Assistants work as partners with humans. Agents are autonomous and once given a task or project do not need further human input before completion. Present GPTs should probably be classified as assistants, but agents are only the blink of an eye away.

11.3 Improving Traditional Cataloging, Classification, and Retrieval Tools

The elephant in the room here is presented succinctly by Tamar Sadeh in her doctoral thesis and a series of papers including 'From Search to Discovery' (Sadeh 2015). The argument is: the traditional approach required users to learn library systems and articulate the perfect template to launch a search which would then be guaranteed to produce a perfect result straight off. In contrast, the modern user could not care less about library systems. In their daily lives, they use Google search and do online shopping all the time. They enter the information discovery process in a sloppy and haphazard way. But get some, or many, results which are then honed to meet their needs. The process is familiar to them. Sadeh describes this:

> The designers of traditional library information systems, such as library catalogs and databases, were very focused on meeting the needs of librarians and expected that users would invest time and effort in learning how to use the system. The designers of discovery systems, driven by the needs of end users, strive to streamline the end-to-end process of finding and obtaining information and make it as simple and friendly as possible. Rather than offering multiple options to enable users to describe their information need, discovery systems offer users simple search interfaces but complement these with multiple post-search options for assessing findings, refining results, and navigating to other results of possible interest. The look and feel of the interface is similar to that of other information systems that are familiar to users, such as web search engines and online bookstores. Furthermore, recognizing that today's users spend hardly any time reading instructions, developers have made discovery systems very intuitive. (Sadeh 2015, 216)

So, we can look at the topic improving retrieval tools, but some improvements may be for librarians only.

There is the view from Patrick Wilson and Karen Coyle that traditional cataloging theory omits the User from its concerns (Coyle 2016; Wilson 1968; Svenonius 1969). This view invokes *descriptive power*, describing the resource items that libraries have, and *exploitive power* which evaluates and recommends items suitable for patrons or Users on particular occasions. Cataloging does the former, but not the latter to any competent and enthusiastic degree. There is subject classification, but that has not been carried out very well (S. Berman 1971; Frické 2012). There are other library services and tools that help with exploitive power, for example: bibliographies, reference interviews, and similar. But librarianship to date has been weak on exploitive power. Relatively new computer supported search engines, with ranked returns, and 'recommender' systems are strong in these areas. But ML, especially large language models, have the potential to take this to new levels.

Thomas Padilla writes

> ...semantic metadata can be generated from video materials using computer vision; text material description can be enhanced via genre determination or full-text summarization using machine learning; audio material description can be enhanced using speech-to-text transcription; and previously unseen links can be created between research data assets that hold the potential to support unanticipated research questions (Padilla 2019, 12)

This is exactly right. We will supplement this in places and give detail to some of the suggestions.

Over the millennia that librarianship has been practiced, librarians have developed many retrieval tools. Here are a few, supplemented with some modern techniques:

Abstracts
Bibliographies
Book reviews
Catalogs
Citation Indexes
Computer Interfaces
Controlled Vocabularies
Cumulative Indexes
Databases
Dictionaries
Encyclopedias
Finding Aids
Handbooks, manuals, etc.
Indexes
Inventories
Keywords
Nomenclature
Ontologies
Outlines, syllabi, etc
Pathfinders
Reference Interviews
Reference Lists
Registers

Reviews
Search Engines
Single Entry Term, Phrase, or Keyword, Search Boxes
Subject Guides
Summaries
Tables of Contents
Textbooks
Thesauri
Web Browsers

As we will see, ML will likely be able to improve most of these tools individually (see, as examples, (Iris.ai 2023; Pickering et al. 2022)). A different consideration is whether ML can replace some of the tools entirely.

11.3.1 NLP Inspired Improvements

Most of the AI/NLP areas of value to librarianship are mentioned and described in Chapter 9:

- Named Entity Recognition
- Topic Modeling, Text Classification and Automatic 'Tagging'
- Controlled Vocabularies, Thesauri, and Ontological Vocabularies
- Automatic Indexing
- Text Abstracts, Extracts, Key Phrases, Keywords, and Summaries
- Sentiment Analysis

- Author and Genre Recognition and Plagiarism Detection
- Text Mining and Question Answering
- Machine Translation

In general terms, these all make improvements to older techniques. But some bring features that are genuinely new:

- Topic Modeling can identify new topics in huge corpuses of texts e.g. in social media.
- Indexing can identify sources outside of the original indexed document. This could be useful for research.
- Sentiment Analysis can be useful for recommender systems. For individual books, for authors, for genres. For individuals, for the patrons as a whole ('trending books') and for collection development.
- Author and Genre Recognition and Plagiarism Detection.
- Text Mining and Question Answering.
- Machine Translation.

11.3.2 Metadata Generation and Automatic Cataloging

Machine learning can certainly be used to create all the forms of metadata that are in use in librarianship today i.e. it can do cataloging. (See for example (Griffey 2019; Corrado 2021)). What would be likely here is interactive reinforcement learning or human-in-the-loop learning. That is, during the training process professional catalogers would provide feedback

as to how well the automatic system is performing. The catalogers would be part of the training. (Of course, it is not easy to know what good cataloging amounts to (Snow 2017).)

In certain circumstances, automatic metadata generation has the potential to be very useful. For example, UNESCO has audio recordings that would benefit from having metadata. They have about 6500 recordings, in 70 languages, and some of the recordings are multilingual. What they have done in the past is that an intern listens to a recording and picks topics and personalities. There are ML systems that can do the speech recognition and transcription (for example, Whisper from Open AI (OpenAI 2022b)). Apparently one of the challenges these systems can have is with crosstalk (e.g. meetings where several different people talk as and when they feel so inclined).

11.3.3 Some Retrieval Tools

The opportunities or possibilities here are extensive. We will address just a few examples.

Producing a *List of References* that are actually cited in a text is a trivial computer science problem. There are many human-powered citation managers (e.g. EndNote, Zotero, and Microsoft Word), and all of these can produce 'Bibliographies' (here meaning reference lists, or citation lists). True *Bibliographies* are another matter. A 'True Bibliography' is being

understood in this context as being a list of the works used to write or produce the text, whether *those works are cited or not*.

Producing a true *Bibliography* is no easy task. The author or authors of a work can do it. But for third party humans or computer programs, it is a challenge. Computational linguistics can classify texts, can identity the genre, can identify whether two works were written by the same author, can identify plagiarism, etc. But these methods, whether they use ML or not, mostly rely on what is in the text explicitly. Locating what is in the background that might have inspired the explicit text is a challenge of a different level entirely. As of 2023, it cannot be done.

Paula Carina de Araújo and fellow authors and, separately, Linda Smith, provide a good introduction to the vast area of *Citation Analysis* and *Citation Indexes* (Araújo, Castanha, and Hjørland 2021; L. C. Smith 1981). In brief, a reference is an explicit acknowledgement that one text gives to another, and a citation is reference received by a text from another. Citations and references can be used to build webs or networks between documents. Such networks, which often amount to citation indexes, can be valuable for a variety of scholarly purposes. Centrally

> … (1) they are tools for the scholars seeking knowledge and (2) they are tools for the scholars studying science (including scientometricians and information scientists). (Araújo, Castanha, and Hjørland 2021)

One point to note about traditional citation analysis is that what is being considered is actual citations made by human authors. These citations may

be made for a variety of reasons. (As Lizzie Gadd notes, the *Citation Typing Ontology* lists 43 different types of citation (Gadd 2020; Peroni and Shotton 2012).) Let us pick a semi-random five of these: *is confirmed by*, *corrects*, *critiques*, *derides,* and *disagrees with*. Consider a particular research paper, A, and another research paper B. If the human author of A is conscientious and knowledgeable, she may cite B for any of the five, or other, reasons. But, nevertheless, given the vagaries of life, she may not cite B at all. However, the paper A may still be confirmed by B (or correct B or critique B etc.). That is what the paper may do, objectively. A scholar of ideas, of intellectual history, of the development of the theories in a field, may mainly be interested in the relationships between A and B, not in whether the author of A cited B. Additionally, consider the time before it was the practice to make citations. Ancient Greek or Roman authors wrote texts that, for example, critiqued other texts. Human creators of citation indexes or analyses basically can only work with actual citations. But at this point ML and NLP systems have a crucial advantage. They can scan the entire research literature and form a knowledge or information map of which papers confirm which other papers (or deride which other papers, etc.). They can scan the content (as well as the gossip of actual citations). New knowledge mapping tools are, and will be, far superior to their traditional counterparts (see, for example, (Tay 2022)).

Moving on. ML systems could write *Book Reviews*. With fiction, it could identify plots, characters, themes, whether the content was 'diverse', intended audience, and other aspects of the document or book. With non-fiction, it could assess quality by means of coherence, 'groundedness', truth-and-evidence, writing style, citations it uses, citations to it, and other

indicators. Also, it could, using sentiment analysis and other NLP techniques, collectively assess, summarize, and evaluate reviews written by other agents (human or otherwise).

Libraries make extensive use of *Databases*. Almost all information about their own individual holdings will be held in databases. Access to these will often be via their *Catalog*, which might be in the form of an *Online Public Access Catalog*. Also, there are any number of commercial and other databases that serve as access points to further resources outside an individual library's holdings (for example, to research papers, to legal materials such as citations and precedents). An academic library might provide access to hundreds of outside databases. Of course, databases will be used in the everyday administration and management of libraries (such as for patron and circulation records, for staff salaries, etc.).

Database theory is a specialist area, widely studied in computer science. Databases themselves provide organization to their contents. They also should be able to do so in a structured and provably correct way. They support CRUD operations (Create, Read, Update, and Delete). There is, or can be, plenty of automation in connections with databases, for example, with checking the data on entry (for format, reliability, etc.), checking integrity, following a backup or archiving or compliance policy. It is not clear quite what role machine learning might have in this domain. ML can program a database. It can do any computer programming approaching the level of professional programmers. How good it would be a design is an open question. This is an area where there are many formal techniques— Entity-Relationship diagrams and the like— and ML would presumably

easily master those. Machine learning can have plenty of relevance to the contents of databases, picking up patterns in the data for one reason or another (for example, identifying fraudulent transactions in a financial database). Somewhat similarly in a library setting machine learning will be able to identify usage patterns in the resources— for example, which resources are used by which segments of the patrons— and thus aid acquisitions and collection management.

Traditional *Catalogs* list all the materials held in a library. The lists will have data and metadata about the resources. If a library physically issues books and materials, a catalog might assist with lending, checking availability, placing holds, etc. All of these functions benefit, or will have benefitted, from computer automation. The old timey favorite *Card Catalog* was a technology to help the patrons find items among those materials listed in the Catalog. As computers, networks, and automation came in the Card Catalogs evolved into *Online Public Access Catalogs* (OPACs). OPACs steadily acquired additional functions such as access to materials in other libraries or in other formats like databases. OPACs are probably drifting off into the sunset (Wells 2021). There is a better alternative, the so-called *Discovery Systems* (such as Primo VE, Summon, or EBSCO Discovery Service). Tamar Sadeh writes:

> Discovery systems provide access to a large, diverse information landscape of scholarly materials regardless of where the materials are located, what format they are in, and whether the library owns them or subscribes to them. At the same time, these systems typically offer simple, Google-like searching as the default option, to accommodate the expectations of today's users. With this type of searching, users do not spend much time formulating queries, and their queries often yield large result sets; therefore, discovery

> systems focus on relevance ranking and on tools that help users easily navigate and refine result sets. Librarians have welcomed the advances in discovery services for their users. However, this new reality poses challenges to the practices that librarians have developed over the years and, in some cases, is at odds with the systematic, controlled approach to searching endorsed by librarians (Sadeh 2015).

(with, for example, personalization and recommendation— such as Primo VE, Summon, or EBSCO Discovery Service).

LLMs can create *Dictionaries* — after all, they will have seen massive amounts of text in its natural contexts. Many dictionaries provide examples in use. LLMs would be able to provide richer and more comprehensive examples. Right now, though, 2023, it would be usual to construct dictionaries by editing existing dictionaries. Merriam-Webster's dictionary, for example, is about 200 years old, and about 1000 new words are added each year. Words are removed also. There are corpora— collections of real world text— for example, the Open American National Corpus (anc 2023). Computer analysis of corpora tells which words are new and how frequently they appear, and also which words are drifting out of use. Then, presently, human judgement, assisted by computers, makes decisions on how to edit the dictionary. The whole process could surely be done by ML and LLMs on their own. The LLMs in question would presumably have some downstream training from human experts. The commercial companies— such as Merriam-Webster— do use artificial intelligence, as examples to personalize the experience to the User and to provide usage examples and notes.

There are *Encyclopedias* that have in part, or in whole, been created by ML

- *Wikipedia* uses AI … 'This [use] may be directly involved with creation of text content, or in support roles related to evaluating article quality, adding metadata, or generating images' (Wikipedia 2023p)
- Wikimedia is using ML to help with images for *Wikidata* (Redi 2018)
- *Encylopedia.com* (Encyclopedia.com 2019*)*. This is an access point, rather than an encyclopedia in itself. It gives access to 200 other encyclopedias and can search and summarize.
- Numina Group's *Warehousing 'Encyclopedia'* (NuminaGroup 2023) (This is more of a glossary or catalog.)

ML created encyclopedias could or should be completely up to date, accurate, and comprehensive. They might be expensive, biased, and with some entries that were hallucinations.

A *Pathfinder* is:

> A subject bibliography designed to lead the user through the process of researching a specific topic, or any topic in a given field or discipline, usually in a systematic, step-by-step way, making use of the best finding tools the library has to offer (Reitz 2014).

The finding tools mentioned here include: catalogs, bibliographies, indexes, abstracts, bibliographical databases, and search (by author, title, topic, and keywords).

Machine learning can improve pathfinders in many ways:

- The whole task of producing a pathfinder can be done automatically.
- It can personalize a pathfinder to a User, instead of their being a single pathfinder for many Users.
- Its search will be better.
- It can find and follow topics even when those topics do not have their own metadata by having been catalogued by the library systems.
- Indexes will be better.
- Abstracts will be better.

11.4 Chatbots

Chatbots have been in use in libraries for at least ten years (McNeal and Newyear 2013; Weigert 2020). They are improving. They have the potential to do, or interface to, most of the current tasks where librarians interact with patrons. Most obviously here are Reference Interviews. There may be a loss of some personal touch. But the gain might be a tireless expert reference librarian for every patron, available 24 hours a day 7 days a week. Another possibility is that of supplementing Frequently Asked Questions (FAQs). After all, at the start of a freshman year at a college, there may be several thousand students wanting to ask the same routine questions (such as the opening hours for the campus libraries). Chatbots might also produce a thinning or abandonment of present-day library websites. A deeply structured website will often not be the best way of directing users to the resources on offer (Wikipedia 2022e; Thoppilan et al. 2022).

There may be a loss of personal touch. There may be a gain of personal touch— some folk are really enchanted with chatting with chatbots. There is some evidence that chatbots outside a library setting do not provide a good 'customer experience' (ujet.cx 2022b; 2022a)(Standard point: one benefit might be indefinitely many tireless expert reference librarians, enough for one for every patron, available 24 hours a day 7 days a week.)

Librarians would be valuable, perhaps even necessary, in the creation of suitable chatbots. Likely the chatbots will use the GUS architecture with frames with slots (Bobrow et al. 1977). (The frames provide the contexts, and the slots the values of the variable for the data such as the questions and answers.) Librarians have a better idea than most what the frames should be for a librarianship setting.

11.4.1 Reference Interviews

Some librarians either are reference librarians, or, as part of their duties, conduct reference interviews. Librarians here are acting as intermediaries between patrons or Users and reference or information sources. A few years ago, the aim of a reference interview was to match a patron's needs to a single library's resources. Nowadays the resources would be assumed to extend outside a single library perhaps to all libraries and, indeed, to the Internet itself.

Several sources provide the same instructional guide to librarians as to how to conduct a successful reference interview. Here is a typical outline:

Purpose:

Allows staff to match the customer's question to a relevant and useful source of information. The aim of the interview is to answer the patron's questions using the library's resources.

Guide To A Successful Reference Interview:

1. Approachability
2. Interest
3. Listening/Inquiring
4. Searching
5. Follow-up

1. Approachability

Pay attention to both your own and the customer's body language. Acknowledge and greet the customer as they approach the desk. Ensure the customer has your full attention.

2. Interest

Maintain eye contact
Find a confidential location for the customer to ask a question
Restate and rephrase the question
Speak in a relaxed tone
Make the customer feel comfortable
Nod your head when the customer starts to ask questions

3. Listening/Inquiring

Do not interrupt
Ask clarifying questions
Let the customer express their needs in their own words
Ask open-ended questions to probe about their information needs
Examples:

Tell me more about the sources of information you already consulted?
Why do you need the information?
How will you use the information?

Remember, WORF

Welcoming, Listen Carefully
Open-Ended Questions

Repeat their answer back to the customer
Follow-up to ensure they've found the information

4. Searching
    Keep the customer informed of the progress
    Offer referrals
    Offer to instruct the customer on how
    Ask clarifying questions:
        Do you want printed information that you can take home with you?
        Do you have access to a computer so you can look up sources online?
5. Follow-up
    Asking the customer if they have everything they need ensures that the customer is satisfied with the transaction.
    If the follow-up questions indicate that the customer is not satisfied:
        Clarify what information is missing
        Offer to continue working on answering the question
        Refer the customer to another organization if material is not available at your library

[The author dislikes the use of the word 'customer' but maybe that is just him.]

ML and LLMs can certainly excel at all this. Joseph Vincze has a useful discussion in his paper 'Virtual Reference Librarians (Chatbots)' (Vincze 2017). [A caution: that paper was written before the advent of ChatGPT.]

11.4.2 Virtual Services

There have been Virtual Reference Desks, Ask-a-Librarian web pages, and Chat-with-a-librarian through a web page, more-or-less since the internet expanded and became popular. Some of these have been retired, for

example the Library of Congress Virtual Reference Desk was retired around 2020.

The LLMs complete change the game on this. You can have ChatGPT on your smartphone. That is a virtual reference desk, without needing a library or librarians. As of June 2023, ChatGPT is not perfect with accuracy, providing references, and avoiding hallucinations. But it is improving all the time, and there are good reasons to suppose that ChatGPT with appropriate plugins will outperform any extant virtual reference desks. Librarians might work with ML researchers to create the plugins.

11.4.3 Chatbots as Continuous User Testing of a Library's Public Interface.

This might be controversial, and it certainly would need handling carefully. But chatbot transcripts would throw good light on what patrons actually do, or plan to do, while using a library's resources. For example, a web page with a large number of show/hide toggles (accordion widgets) may prove hard to follow; a chatbot leading a user through the page could provide feedback on this.

11.5 Release, Produce, or Curate Training Data

ML is only as good as its data. Training data is hard to come by. It needs to be plentiful, and it needs to be of high quality. For supervised learning, the

data needs to be labeled accurately. Librarians are well placed here with all kinds of suitable data. They already have rich metadata on traditional resources, including shelf classification, subject classification, and indexes. And they can produce new kinds of data. For example, the READ-COOP projects to use handwriting recognition on archives of historical documents (READ-COOP 2021). These projects can involve creating and inspiring crowdsourcing to produce data at scale, e.g.

> Transcribe Bentham is an award-winning participatory initiative which launched in 2010. Its aim is to engage the public in the online transcription of original and unstudied manuscripts written by Jeremy Bentham, his correspondents, and his amanuenses. (UCL 2018)

(For an explanation of crowdsourcing see (Wikipedia 2023c).)

Public librarians can inspire crowdsourcing. Many modern ML projects, especially Large Language Models, and Foundation Models, use crowdsourcing in their training. For this they generally need 'ordinary' people, but collectively the crowds would usually need to be diverse (as to race, ethnicity, religious beliefs, sexual orientation, etc.). Public libraries interface with many hundreds of thousands of exactly the kinds of folk that would be suitable, possibly even ideal.

11.6 Debunking, Disinformation, Misinformation, and Fakes

Librarians have always been active with information literacy, which is helping users and patrons to become more discerning and skillful in their approaches to information resources. (We will discuss this again later.) But, in addition to this more general raising of skills on the part of users there is or can be a scrutiny of the actual resources themselves. There are problems aplenty nowadays with disinformation, misinformation, fake news, deep fakes and the like (Meszaros and Goodsett 2022). Traditionally, librarianship would not pass a view on the veracity of sources or on evidence. Librarians would remain neutral, for example, between resources on evolutionism and on creationism (and there are good free speech arguments for doing this). But there is considerable value now in fact checking.

Librarians already help patrons to fact check e.g. (Knapp 2021). Many libraries and library associations are deeply involved in fact checking Really this is a part of epistemology, perhaps social epistemology.

11.7 Social Epistemology

Margaret Egan and Jesse Shera introduced social epistemology as being:

> ... the analysis of the production, distribution, and utilization of intellectual products in much the same fashion as that in which the production, distribution, and utilization of material products have long been investigated.(Egan and Shera 1952)

One core part of this concerns what we know— the true beliefs we have—as individuals and collectively. Most of what we know individually comes from other people via recorded knowledge. Certain practices regarding that recorded information can promote, or inhibit, social knowledge (i.e. knowledge aggregated across individuals). As an obvious example, easy wide access to recorded information promotes social knowledge whereas censorship inhibits it. Some qualifications are needed to this example. Too much information might overwhelm our attention and interest. A little censorship, or filtering, or curating, might highlight the pearls among the dark sea of many biased and unsupported opinions. Egan and Shera, and other later researchers, such as Don Fallis, Steve Fuller, and Alvin Goldman, have seen the promotion of social knowledge as being a vital function of librarianship (Fallis 2006; 2002; Fuller 1988; Goldman 1999; Egan and Shera 1952).

Librarians are experts at traditional information acquisition, and information provision practices. That is a good start. But machine learning is both going to provide more powerful tools to help with social knowledge and, perhaps a mixed blessing here, a vast amount more source material for those tools to be used on. Here is a conjecture about recorded text, especially new materials on the Internet and on Social Media. The Large Language Models, such as ChatGPT, Bard, and Bing, can write English, and other languages, as well as native speakers. They can do so quickly, much quicker than native writers, and cheaply, much cheaper than native writers. Very shortly, tools from these models will be in the hands of anyone that

wants them (as tools on the web or in word processors or stand-alone apps on smartphones, tablets, or computers). What the tools will produce will be plausible in terms of vocabulary and grammar. Some of the textual products will be information. Others will be misinformation. Some will be 'hallucinations'. Some will be fiction, intending to be fiction. Some will be fiction, not intending to be fiction. It seems that ML source creators will be as ubiquitous as spell checkers are today— every means of producing content will have available an LLM assistant.

What might be roles for librarians in connection with ML and social epistemology? Here are some possibilities:

- Fact checking. Help with identifying misinformation, disinformation and 'false facts'.
- Help with cognitive biases. Many folk have trouble with reasoning— with hypotheses, evidence and truth. Indeed, one experiment showed that 50% of Harvard Physicians can commit the base-rate fallacy. (The base-rate fallacy is explained in Appendix C.2.) AI can help keep people on track. AI can construct proof trees from arguments and evidence. There are other relevant cognitive biases. For example, the phenomenon of confirmation bias suggests that almost everyone mistakenly favors evidence that supports their views, downplaying or ignoring disconfirming evidence. ML and LLMs do not cause this, but they can be used to counter-act it. For example, for patrons interested in balanced view of, say, climate change, librarians, using LLMs could find the confirming and disconfirming evidence. The librarian’s role here would be part synergist, part sentry, and part educator.

- There is the notion of Veritism, or truth-centered epistemology (Kitcher 2002; Nawar 2021), and from this the question arises of whether social epistemology needs to be veritistic. There is the need for input from philosophy. Once there is an acceptable answer to this, ML may be able to help.
- There is the view, now becoming widespread, that Peer Review as used, for example, in scholarly publishing has failed. One proponent of this is Adam Mastroianni (Econtalk 2023). ML can do everything that peer review is supposed to do: check spelling, grammar, citations, diagrams and figures, calculations, originality, contribution to the research field, absence of plagiarism, etc.

11.8 Robots

For convenience here, we will divide robots into three categories: chatbots, humanoid robots, and non-humanoid robots. [There may be some overlap of these boundaries; for example, a humanoid robot might have chatbot capabilities.] Then, orthogonal to this, a physical library may be *using* robots, or providing *access to* robots.

Chatbots are discussed elsewhere (and there are many opportunities for librarians with chatbots). Humanoid robots are beginning to be introduced as companions and helpers to the elderly in rest homes. There are also some uses in nursing. Such robots are often mobile and can converse. The 'humanoid' part usually includes humanlike expressions of emotions and gestures. These can engender trust and reduce anxiety on the part of the

people the robots are interacting with. Somewhat similarly to being helpers in rest home, robots have been trialed in public libraries. As examples of actual uses, being a teller of stories in story-telling sessions for children, being a greeter to the library and answering directional or locational questions to books or resources (Nguyen 2020; Kim 2017). Non-humanoid robots, for example, welding robots used in the manufacture of cars, would usually appear as part of the automation of processes. There has been inventory control in general, using RFID (Radio Frequency Identification) labels and chips, since the 1980s. This means that it is relatively easy to automate closed stack systems (where the public do not browse the books on the shelves). The physical collection can be 'in the basement' and automation will do the rest. RFID, and similar technologies, are also invaluable with open stack systems. A hand-held scanner, or a wand, or even built in systems, in the shelves or walls of the building, will find, for example, any book or identify that it is missing. Increasing use of automation may be useful, but there does not seem to be a large role for robots. As to libraries providing access to robots, robots are going to be an important part of our future. This suggests that librarians can help educate the populace by, for example, lending robots or having makerspaces with access to robots or having educational seminars on robots.

In sum. It is unclear quite what might happen in general with humanoid and non-humanoid robots in libraries, and what the opportunities might be. (See also (Tella 2020; Tella and Ajani 2022).)

11.9 Images

A considerable portion of ML concerns imaging or has images as its subject matter. There are various librarianship related problems here e.g. recognizing or classifying images, attaching metadata to images, searching among a collection of images, perhaps where the input is itself an image, and so forth. Images are reasonably important within librarianship. There are libraries with images as part of their collections. But also, many text sources, for example newspapers, have images or diagrams within their content. Additionally, there is film or video.

Such sources can be digitized, and from there processed using ML for certain tasks. It all becomes a matter of numbers and patterns in numbers. Some modern foundation models are, or aspire to be, 'multi-modal'. This means that they can work with source and output in any medium (e.g. text, or images, etc.)

An example is Benjamin Lee et al. *The Newspaper Navigator Dataset: Extracting And Analyzing Visual Content from 16 Million Historic Newspaper Pages in Chronicling America* (Lee et al. 2020). They write in their abstract:

> … we introduce a visual content recognition model trained on bounding box annotations of photographs, illustrations, maps, comics, and editorial cartoons collected as part of the Library of Congress's Beyond Words crowdsourcing initiative and augmented with additional annotations including those of headlines and advertisements. We describe our pipeline that utilizes this deep learning model to extract 7 classes of visual

content: headlines, photographs, illustrations, maps, comics, editorial cartoons, and advertisements, complete with textual content such as captions derived from the METS/ALTO OCR, as well as image embeddings for fast image similarity querying.

We have here machine learning, crowdsourcing, additional annotations… on 16 million pages. Respect!

11.10 Annotated Readings for Chapter 11

Sanji, Majideh, Hassan Behzadi, and Gisu Gomroki. “Chatbot: An Intelligent Tool for Libraries.” Library Hi Tech News ahead-of-print (2022). https://doi.org/10.1108/LHTN-01-2021-0002. (Sanji, Behzadi, and Gomroki 2022).

Tay, Aaron. “List of Innovative Literature Mapping Tools | Aaron Tay’s Musings about Librarianship,” 2022. https://musingsaboutlibrarianship.blogspot.com/p/list-of-innovative-literature-mapping.html?view=classic. (Tay 2022).

Apparently there have been robots in libraries for some time. Here, from 1965, is Lawrence Lipton demonstrating his 'robot' Duhab (Detector of Undesirable Habitués). Duhab finds undesirables such as censors and book burners.



Figure 27. Duhab (https://digital.library.ucla.edu/catalog/ark:/21198/zz0002tw94 https://en.wikipedia.org/wiki/File:Lawrence_Lipton_and_DUHAB.jpg).

Chapter 12: Librarians as Sentries

12.1 Copyright and Intellectual Property

> As librarians license access to content from vendors, we need to ensure that contracts do not preclude our users from conducting text and data-mining research, algorithmically based research, and machine learning. (Miller 2020)

12.2 Intellectual Freedom

ChatGPT tells us that intellectual freedom has several drawbacks, including:

- Misinformation and false ideas: People may spread false or harmful ideas that are protected under the umbrella of intellectual freedom.
- Conflict with other values: Intellectual freedom may conflict with other values such as privacy, security, and safety, leading to difficult ethical dilemmas.
- Limits to freedom: In some cases, laws and regulations may limit intellectual freedom in order to protect public interest or prevent harm.
- Divide between different perspectives: Intellectual freedom may lead to a fragmentation of perspectives and ideas, making it more difficult for society to reach consensus on important issues.

- Resistance to change: Some individuals and groups may resist new ideas and perspectives, leading to resistance to social and cultural change.

[OpenAI . (2023) This paragraph was written with ChatGPT.]

12.3 Censorship and Algorithmic Curation

Librarians can defend against censorship. ML is not directly a censor, but many of the techniques it enables and improves— such as personalization, recommendation, search— can lead to inadvertent censorship. The techniques can lead to filters and to placing individual patrons within their own bubbles, or silos, of information to the exclusion of the wider world of information. Customization is good and will presumably be appropriate and well received by patrons. But censorship is bad. There needs to be a balance. Removing inappropriate and offensive content entirely can be good, but it is hard to draw the boundary lines. Social media are also present. One would not ordinarily think extensively about social media in the context of libraries, but social media is a gatekeeper to some, perhaps even most, of the information that students and patrons obtain. There is also the 'manipulative capabilities of algorithmic processes'— the abilities of algorithmic processes to influence a person's thought, emotions, and decisions (IFLA 2020, 5). In some ways this is not a lot different to advertising or propaganda. But ML is more powerful and less obvious and transparent. Librarians might have a role as sentries here. They have a history of protecting patrons, and to arm wrestling with censorship.

Some AI tools improve access to information, but there is a need to check that the improvements are available to all including to people with disabilities.

There is also inadvertent censorship. ChatGPT tells us:

Inadvertent censorship refers to the unintended restriction of information or expression due to various factors. Here are some examples:

- Technical difficulties: Technical glitches or failures in communication systems can unintentionally restrict access to information or expression.
- Overzealous filtering: Some internet filtering systems, designed to protect users from harmful content, can inadvertently censor legitimate information or expression.
- Misapplication of laws: Laws and regulations designed to protect national security, public order, or public morals can be misapplied, leading to the unintentional restriction of information or expression.
- Economic considerations: Economic factors, such as the cost of publishing or distributing materials, can lead to the unintentional restriction of information or expression.
- Social norms: Social norms and cultural biases can lead to the unintentional restriction of information or expression that goes against the prevailing norms and biases.

[OpenAI . (2023) This paragraph was written with ChatGPT.]

12.4 Privacy

Centrally, this concerns Users' behavior and patron data. Privacy is a core value of librarianship (American Library Association 2006; E. Berman 2018). The main justification for this is to allow patrons intellectual freedom.

> Lack of privacy and confidentiality has a chilling effect on users' selection, access to, and use of library resources (American Library Association 2006).

Modern computerized libraries typically use Integrated Library Systems (ILS) (sometimes known as Library Management Systems (LMS)) to manage their services. These systems collect data, some of it about individual patrons (e.g. name, age, address, email, phone, driver's license, gender, borrowing history, etc.) (E. Berman 2018). Outside agencies, for example credit reporting, and credit card, companies also collect data about individuals. It is possible to collate data across different systems, for example to use library patron data in conjunction with credit card data. Collation like this— data collected for one purpose being used without consent for another— should never happen. But it would be wishful thinking to suppose that it never does. There are issues here of privacy and consent.

Librarians might be watchful that ML applications meet the required legal and ethical standard.

12.5 Bias

ML systems can be biased, and often will be if they are LLMs pre-trained by self-supervision on large quantities of internet sourced digitized text. Some of these biases can be corrected relatively easily by prompting, fine-tuning, or plugins. But it is helpful to all if it is known what the biases are. Librarians can be on guard to ensure that their services are fair, equitable, and do not commit harms of representation or omission. As mentioned earlier, librarians do manage bias in many aspects of their work, for example with collection development. Most of these traditional forms of potential biases are nothing to do with machine learning, nor have been caused by machine learning. However, ML may be able to help with redressing them.

12.6 Social Epistemology

12.6.1 Reliability, Validity, and Over Confidence

Most machine learning systems can make mistakes. Generally, they work in terms of probabilities and will not give the same answer twice. This means that there is a problem with reliability. Further, if the different answers contradict or are inconsistent with each other, some answers must be plain wrong. This means that there is a problem with validity. In research in general, if the researcher has an instrument that is unreliable and lacking in

validity, typically that instrument would just be discarded. The probabilities here also concealed. An LLM would not ordinarily say 'there is a 70% chance of rain and a 30% chance of sunshine'. Rather, it would say definitively 'it will rain!' about 70% of the time, and, definitively, 'there will be sunshine!' about 30% of the time. The LLM displays an exaggerated confidence that a user might have difficulty in assessing and understanding. Librarians can help patrons with attitudes to information that ML systems provide.

12.6.2 Confirmation Bias and Poor Reasoning

As mentioned in Section 11.7, librarians, assisted by AI, can pay attention to cognitive biases.

12.6.3 Misinformation

Librarians can guard against misinformation. They can ensure that the information provided to patrons is accurate, reliable, and up to date.

12.6.4 Awareness of the Digital Literacy of Patrons

Librarians can have a sense of the information and digital literacy among the community of patrons in their libraries. If there are shortcomings, the

librarians can provide further guidance, education, and support, especially with the question of how ML impacts the information patrons receive.

Going forward, ML is going to provide many systems and tools for librarianship. Librarians themselves can evaluate these tools to ensure they meet the needs of their patrons and the communities.

12.7 Chatbots

An eye needs to be kept on chatbots. There is the well-known historical example of Microsoft's Tay from 2016 (Wikipedia 2023n). Once in the wild and available to all, this chatbot degenerated into sexist and racist garbage within a day. This is an old example now. Also, it was configured to learn from the people it was talking to, and bad actors set out to lead it astray. They succeeded.

Chatbots can have several drawbacks, including:

- Limited understanding: Chatbots are limited in their understanding of human language and context, leading to misunderstandings and inaccurate responses.
- Lack of emotions: Chatbots lack emotions and empathy, making it difficult for them to fully understand human behavior and respond in a meaningful way.

- No creativity: Chatbots lack creativity and imagination, which can limit their ability to provide innovative solutions to complex problems.
- Inflexibility: Chatbots are designed to follow specific scripts and rules, making it difficult for them to adapt to changing circumstances or new information.
- Reliance on data: Chatbots rely on data and algorithms to generate responses, which can perpetuate biases and perpetuate harmful information.
- Security risks: Chatbots can be vulnerable to hacking and other forms of cyber-attacks, which can compromise sensitive information and put users at risk.
- Technical limitations: Chatbots can be limited by technical constraints, such as limited processing power, storage capacity, and connectivity.

[OpenAI . (2023) This paragraph was written with ChatGPT.]

12.8 Personalization and Paternalism

Search usually will rank the links it returns. If there is ranking, or personalization, there is further opportunity for error or malfeasance. Most obviously the finger is on the scales with systems that allow sites to pay the search providers for higher positions in the ranked returns. ML systems need to be scrutinized in these areas.

Paternalism ‘is to act for the good of another person without that person’s consent, as parents do for children’ (Suber 1999). Generally speaking, outside of concerns of machine learning:

- parents being paternalistic to their own children is fine,
- some adults, such as teachers, being paternalistic to other peoples’ children can be fine,
- some adults, such as doctors and nurses, being paternalistic to adults can be fine,
- otherwise, adults being paternalistic to other full competent adults is often suspect.

There is an extensive literature on paternalism in librarianship, especially in so far as that might apply to children. (See, for example, (Frické, Mathiesen, and Fallis 2000).) This is a complex area. But there is a statement from John N. Berry that gets to the nub of the issue:

> Nor must we deprive [children] of the nurture, the helping hand, the guidance, the tools for seeking truth and knowing when it is discovered. We cannot simply turn them loose in our jaded information society without helping them understand that some of the information is false, is evil, is dangerous, is misleading, or is ambiguous… That may not be a legal obligation, but it is clearly a moral duty for every librarian, every teacher, every parent and person in a free society (Berry 1998).

(Berry wrote this in 1998, long before LLMs and a powerful internet.)

The present text is primarily on research libraries and advancing knowledge. Children are not front and center here. Nevertheless, LLMs have the potential to act for good and to cause harm, both to children and to adults. Attention should be paid to questions of when paternalism is appropriate and when it should be avoided. Helping folk understand that some ‘information’ from LLMs is false, evil, dangerous, misleading, or ambiguous is a good idea, even if it might amount to paternalism.

12.9 Images and Facial Recognition Technology

Attaching metadata to images is going to have its adventures. It is going to make mistakes. There are going to be false positives and false negatives. What turns on this is an open question. A notorious example is from 2015 when Google Photos misclassified Jacky Alciné and his friend, who are both black, as being gorillas. It is not really known why this happened, nor, it seems, has it been fixed (Grant and Hill 2023). The author has had a mildly similar experience himself. In 2016, Richard Lee, a New Zealand man of Asian descent, had his passport photograph rejected by software because 'his eyes were closed'. Lee, an engineering student, told Reuters 'No hard feelings on my part, I’ve always had very small eyes and facial recognition technology is relatively new and unsophisticated' (Reuters 2016). (Give him a medal for saying that.) The present author is a New Zealander, and around 2016 he tried to renew his New Zealand passport and the passport robot rejected his photographs because 'his eyes were closed'. A couple of matchsticks and Photoshop fixed that!

Attention does need to be paid to the attachment of metadata to images.

Facial recognition technology in a general setting is discussed earlier in Section 7.7.3 (Buolamwini et al. 2020; American Library Association 2018). Our guess is that it will be used in libraries, without controversy, wherever and whenever patrons need to establish their identities e.g. when borrowing books. Tracking patrons is another matter. Attention needs to be paid to it.

12.10 Losing Jobs

There is a standard answer to questions of automation causing loss of jobs. It is: automation automates routine repetitive jobs thus freeing up workers to do more complex and valuable tasks. Of course, this does not mean that there will be no loss of jobs. There is a standard text on this question. It is Kenning Arlitsch and Bruce Newell's *Thriving in the Age of Accelerations: A Brief Look at the Societal Effects of Artificial Intelligence and the Opportunities for Libraries* (Arlitsch and Newell 2017). It is slightly old in that since 2017 LLMs have opened the floodgates on what AI can do. Daron Acemoglu has a briefing *Get Ready for the Great AI Disappointment* which argues that the effects of LLMs in the near term will prove to be 'so-so automation' which perhaps displaces workers but without large gains in productivity (Acemoglu 2024).

12.11 Annotated Readings for Chapter 12

Acemoglu, Daron. “Get Ready for the Great AI Disappointment.” Wired, 2024. https://www.wired.com/story/get-ready-for-the-great-ai-disappointment/. (Acemoglu 2024)

Arlitsch, Kenning, and Bruce Newell. “Thriving in the Age of Accelerations: A Brief Look at the Societal Effects of Artificial Intelligence and the Opportunities for Libraries.” Journal of Library Administration 57, no. 7 (2017): 789–98. https://doi.org/10.1080/01930826.2017.1362912. (Arlitsch and Newell 2017) This has useful material on the question of whether library jobs will be lost to automation, and, if so, which positions.

Cook, John. “Cranky Uncle.” Cranky Uncle, 2023. https://crankyuncle.com/. (Cook 2023) One way of combatting misinformation.

Tait, Elizabeth, and Cameron M Pierson. “Artificial Intelligence and Robots in Libraries: Opportunities in LIS Curriculum for Preparing the Librarians of Tomorrow.” Journal of the Australian Library and Information Association 71, no. 3 (2022): 256–74. https://doi.org/10.1080/24750158.2022.2081111. (Tait and Pierson 2022)

Chapter 13: Librarians as Educators

13.1 Information Literacy (for Consumers of Information)

Information literacy in the age of AI is a new beast. As always, there is the helping patrons and users navigate the interface between information and their good selves. But there is a lot more nowadays, now we have machine learning, large language models, and algorithmic pipelines that might use biased data and might produce biased results.

13.2 Artificial Intelligence Literacy

> What does ‘AI literacy' entail? Alongside basic digital literacy and [Information Technology] skills, ‘AI literacy’ usually begins with an **elementary understanding of how Artificial Intelligence and Machine Learning work**, what they can and cannot do (IFLA 2020, 10 their emphasis).

Michael Ridley and Danica Pawlick-Potts write:

> Navigating the effects of AI as well as utilizing it in a responsible way requires a level of awareness, understanding, and skill that is not provided by current digital literacy or information literacy regimes (Ridley and Pawlick-Potts 2021a).

Others express similar views. For example, Charlie Harper writes:

> The privacy (or really lack thereof) and ethics of data collection and dissemination should become an integral part of information literacy services. Now that images, video, and audio can be faked

> in staggering ways, the reality of source origin is becoming increasingly messy, too. Facilitating and promoting critical thinking and awareness within the community is a must (Harper 2018).

Librarians have the opportunity to help the population with this new kind of information literacy. There are some new labels that are appearing here, for example, 'AI literacy', 'Digital literacy', 'Computer literacy', 'Machine Learning literacy' and even 'Algorithmic literacy' (Ridley and Pawlick-Potts 2021a; Digital2030 2022; Druga et al. 2019a; Carlson and Johnston 2015). We will come back to these names shortly. These literacies have two classes of potential students: library staff, and the patrons of libraries (i.e. the public at large, or students and researchers in colleges and universities). Then we may make a distinction between 'internal' (which here means 'intellectual') and 'external' (which here means 'social', understanding 'social' in a wide sense). So, for example, a person has AI literacy at an internal level if they understand many of the kinds of AI software programs and possibly could even write them. Roughly, this would be college level knowledge (or 'competencies') that include some AI or computer science courses. AI literacy at an external level requires knowledge of the settings or societal contexts in which AI is used, for good or for bad, and a reasonable grasp of how AI affects their own lives and that of societies as a whole. Even children can be AI literate, at an external level.

Back to the labels. We favor 'AI literacy'. ML is where the action is, but 'Machine Learning literacy' is a mouthful. Neither 'Digital' nor 'Computer literacies' are entirely accurate. 'Algorithmic literacy' is unfortunate. It smears the word 'algorithm' into a meaning it does not have, and then

exhorts us to be literate about a topic that it does not itself define correctly. Nevertheless, no doubt 'Algorithmic literacy' will gain ascendance in the marketplace for labels. At which point we will go with the flow. Anyway, it is the *external* versions of the literacies that librarians need to learn themselves and teach to others.

There is a literature on teaching machine learning. For example, the delightfully titled 'Can You Teach Me To Machine Learn?' (Sulmont, Patitsas, and Cooperstock 2019). See also (Nori et al. [2019] 2023; Druga et al. 2019a). The literature, though, is sparse, and the domain unexplored. It is not really known how to do it well. For librarians and libraries, probably hands on workshops, tutorials, and coding classes would be a good idea.

We have dealt with many of the components of AI literacy elsewhere, under different headings. But just to list many of the topics

- Algorithms and how they work
- Having a critical understanding of AI tools (for example, the ones mentioned in Chapter 5) and the information that they may provide
- Bias
- Privacy. Teaching people to understand how to protect their privacy in the context of machine learning is important. Also important is understanding the privacy policies of ML companies and ML applications.
- Facial recognition technology
- Research guidance Librarians could guide researchers in using machine learning tools to analyze their data. This could include

providing advice on the appropriate algorithms to use, helping to interpret the results, and ensuring that the research is conducted ethically.

- Social epistemology

13.3 Data Information Literacy (for Producers of Information)

There is the topic of 'Data Information Literacy' (Data Information Literacy Project 2023; Carlson and Johnston 2015). Jake Carlson and Lisa Johnston ask:

> … what data management and curation skills are needed by future scientists to fulfill their professional responsibilities and take advantage of collaborative research opportunities in e-science and technology-driven research environments? …. how can academic librarians apply their expertise in information retrieval, organization, dissemination, and preservation to teaching these competencies to students? (Carlson and Johnston 2015, 2)

Roughly, researchers need to curate their data in such a way that the data is available to other researchers and society at large. Also, for example, funding sources such as the National Science Foundation require a Data Management Plan for research that they support. This is an ideal area for cooperation between librarians and researchers. This time the librarians are helping the researchers as producers of information, not as users of information. Only a part of this general area involves, or might involve, machine learning. But some of it may well do. Lisa Johnston and Jon Jeffryes discuss a case where civil engineering students put sensors on

bridges to learn of the integrity of the bridges (Johnston and Jeffryes 2015). That challenge may well involve machine learning. As noted earlier, machine learning data, say for supervised learning, needs to be of a certain kind and style.

13.4 Changes in Learning and Teaching

Teaching and learning are changing in colleges and universities, particularly in areas like law and business. If curricula change, then there will need to be changes in libraries. Likely learning will become more personalized. If so, there will be learning data and analytics both of individual students, of instructors, and of groups and classes. Library use data will be part of this. There are initiatives on the use of AI in Teaching and Learning (see, for example, (Office of Educational Technology 2023))

13.5 Scholarly Communication

Jason Priem argues in his paper 'Beyond the paper' that:

> The journal and article are being superseded by algorithms that filter, rate and disseminate scholarship as it happens. (Priem 2013)

This is an interesting and important paper. Priem, writing in 2013, locates many of the changes and possibilities for change in the Web and what the Web enables (for example, share early share often, filter, and crowdsource review). We could not agree more with the general thesis, but now we

would locate the causal factors more with machine learning and LLMs. AI has the potential to remove peer review and possibly even many traditional journals.

13.6 Academic Libraries Collaborating with other University Units

Academic libraries do this already. But ML is going to bring about big changes in teaching and research.

The International Federation of Library Associations and Institutions (IFLA)'s 2020 Statement on Libraries and Artificial Intelligence (IFLA 2020, 14) mentions as example collaborations Stanford University Library AI Studio, the University of Rhode Island's AI lab in the University Library, and the University of Cincinnati Libraries' Digital Scholarship Center (see also (McKenzie 2018)).

A surprise is how few collaborations there seem to be. Most colleges or Universities have a Data Science department or school. All Universities have their libraries. Yet the visible collaborations can be counted on the fingers of one hand.

13.7 AI Laboratories in the Library

In 2017-8 the University of Rhode Island placed an AI lab in its library:

> The library, as an interdisciplinary space that values inclusivity, is the ideal place for people of all backgrounds to learn about AI [. . .] Unlike a typical AI lab focused on research, the URI AI Lab will offer students and instructors the chance to learn new computing skills, and also encourage them to deepen their understanding of AI and how it might affect their lives, through a series of talks and workshops. The lab will offer beginner- to advanced-level tutorials in areas such as robotics, natural language processing, smart cities, smart homes, the internet of things, and big data (McKenzie 2018).
>
> Putting the lab in the University's library is strategic. Organizers hope that students majoring in different fields, from philosophy and ethics to computer science and biomedical engineering, will visit the lab and use it to brainstorm about important social and ethical issues today and create cutting-edge projects (Rhody Today 2017).

The experiences of the AI lab, up until 2022, are described in (Dekker, Ferria, and Mandal 2022). It seems that there has been such a demand for the AI lab, and for Data Science as a whole campus wide, that the University is considering its options.

13.8 Automated Decision-Making

The European Union (EU) has a Generalized Data Protection Regulation (GDPR), (summarized in (Wolford 2018)). It is a law which applies to and protects EU citizens worldwide. It came into effect in 2018. Of particular interest to us is Article 22, which in part reads:

Art. 22 GDPR
Automated individual decision-making, including profiling

1. The data subject shall have the right not to be subject to a decision based solely on automated processing, including profiling, which produces legal effects concerning him or her or similarly significantly affects him or her.
2. Paragraph 1 shall not apply if the decision <further text omitted here> (GDPR 2018)

'Profiling', in GDPR, means:

> … "any form of automated processing of personal data consisting of the use of personal data to evaluate certain personal aspects relating to a natural person"
> Thus profiling should be construed as a subset of processing, under two conditions: the processing is automated, and the processing is for the purposes of evaluation (Goodman and Flaxman 2017).

Article 22, in the context of surrounding text and definitions:

> … restrict[s] automated individual decision-making (that is, algorithms that make decisions based on user-level predictors) which "significantly affect" users. The law … also effectively create[s] a "right to explanation," whereby a user can ask for an explanation of an algorithmic decision that was made about them (Goodman and Flaxman 2017, 1).

European law is European law, of course. But, nevertheless, GDPR is the result of deep consideration and analysis. It highlights the need for caution with use of personal data to produce decisions that affect people and the need for explanations of individual decisions that are made.

Automated Decision-Making (ADM), using data about people as input and producing decisions that affect people as output, has been around for a very long time. For example… The Constitution of the United States requires

that there be a census every 10 years. The results of this census determine how many seats each State has in the House of Representatives (among other things). Processing data on this scale used to be a nightmare. Processing the 1880 census took eight years (Roberts 2019, 100) In 1889, Herman Hollerith received a patent for the 'Art of compiling statistics'. It was a patent for the punched card.

> A hole is thus punched corresponding to person, then a hole according as person is a male or female, another recording whether native or foreign born, another either white or colored, &c. (Hollerith 1889)

The data about each individual person was entered on a single card as punched holes. But then the data could be aggregated by running through the cards and using an electro-mechanical device to detect the relevant holes. The data also could be 'combined' across categories. For example, were you to have an interest in white female carpenters it was essentially trivial in principle to determine how many there were (in a region, or a State, or, indeed, in the United States). Hollerith cards were taken up by wider businesses, and commercial concerns, and the processing techniques led directly to the formation of IBM (and the standard IBM 80 column punched card). Hollerith, and his card, won the 1889 Bureau of the Census competition to mechanize the process of conducting the Census. The 1890 Census was automated.

Thus, there has been ADM for a while, maybe for a hundred and thirty years. You can see the strengths and weaknesses of this original automation of the Census. It makes viable a very difficult practical problem. There

should be few or no errors in the actual processing of the stacks of punched cards. Additionally, the actual processing is completely transparent and, in principle, could be inspected by anyone. But the data needs to be sound. If the data entry for an individual punches a hole for 'farmer' when the individual is actually a 'carpenter', garbage is part of the input and the results of that garbage-in potentially can appear in the output, as garbage-out. Even worse if groups of people are not even included, or properly represented in the census, there certainly could be bias or unfairness in what comes out.

Nowadays, ADM has spread far and wide. Not all such decisions involve data about people as input, not all directly affect people as output. Not all involve machine learning. In fact, probably machine learning forms a relatively small, but increasing, part of ADM. After all, powerful machine learning has been available only for maybe 5-10 years.

Here are some examples where machine learning may place a role in ADM:

- Autopilots on airplanes (and similar technologies on ships, trains, and motor vehicles)
- Many decisions in the financial realm (such as investing, the identification of suspect financial transactions, credit card, or mortgage eligibility (and what interest rates those eligible are entitled to).)
- Medicine (identification of possible positive indicators of skin or breast cancer)
- The behaviors of industrial robots

- Military (robotics on the battlefield including unmanned systems and vehicles of various kinds)
- Providing evidential input to sentencing decisions in criminal trials
- Various decisions involving employment (e.g. interviews, promotions, remedial interventions)
- Making suitable chess moves to be a companionable opponent or to be a tutor of the appropriate level
- And many many more

Some of these types of decision making involve partnerships with humans. Humans may have overrides, may make the final choice or decision, etc.

The strengths and weaknesses here are similar to those of the original Census automation. But there are differences. Many of the systems, or their intellectual aspects, are owned by private companies, sometimes very large companies. A result of this is that often the systems are proprietary and are 'trade secrets'. With this, transparency can disappear. Outsiders may have no way of being assured that the actual processing is sound. When it comes to machine learning, and large language models, for example, even the companies and their researchers can be in a position of not really knowing how their complex systems work (if, indeed, they do work). In this case, transparency is not being hidden by companies, it is simply not available. A factor here is that large language models, for example, are so expensive to establish that only the largest companies or the government have the resources to create them and do research with them. Universities do not have the money to become involved directly. In many fields, one thinks of

universities as being able to be qualified and intelligent honest outside agents. But this is not true for advanced machine learning. The universities cannot participate at all at a core level. Then there is an important difference regarding the data. In immigration controls in airports, and similar, the government uses machine-learning face recognition technology to scan the incoming crowds for 'persons of interest'. The government does not seek anyone's approval or consent to do this. This is similar to gathering the original Census data back in 1890. There is a social contract between members of the community or society that gives implicit consent from members of that society to their government to carry out various governmental tasks and procedures. But the situation with private companies is different. If you walk into a shopping mall and the mall scans your face, without your knowledge or informed consent, how is this right or ethically appropriate? In sum, attention needs to be paid to transparency and the uses of data. There is a lack of insight, a lack of transparency, and often a lack of informed consent. Putting this right perhaps falls to many parties: to the creators of the systems, to the makers of laws, to educators, and to the citizenry at large. Several countries, or groups of countries, do have laws. For example, as we have seen, there is the European GDPR. The USA does not have a relevant all-encompassing laws. Librarians certainly can play a role (Ridley and Pawlick-Potts 2021a). Assume so, what can they do?

The librarians can become knowledgeable themselves. The Masters of Library and Information Science (MLIS) degree, which is the 'union card' for advancement in professional librarianship, might include a courses in the wider aspects of machine learning. (None do, 2023, in the USA.)

Libraries, as institutions, can offer opportunities to their staff to increase their knowledge in this area. This might include courses, workshops, hands on sessions. Then the librarians might be alert to inform their patrons (about surveillance, consent, privacy, automatic decision making, etc.).

13.9 Explainable Artificial Intelligence (XAI)

There is a research field, Explainable Artificial Intelligence (XAI), that has direct relevance to automated decision making. Exposure to the ideas of Explainable Artificial Intelligence (XAI) is of benefit to folk trying to educate others about machine learning.

> Explainable AI is Machine Learning that has the property of being easily understood by humans (Wikipedia 2023e) .

What ML systems can tell us is that certain observable *features*, feature-data, are connected with other observable *labels,* label-data— for example, that being written by J.K. Rowling is correlated with being a popular book. Sometimes an ML system will tell us the labels, the predictions, without being able to tell us of the specific features *that it used in its inference or calculation*. In these cases, the ML system is a black box. You give it a book, the features of a book, and it will tell you if the book will be popular. It is an Oracle. (Henceforward, here, as an abbreviation, 'Oracle MLs'.)

At this point we will take a brief detour into two areas of philosophy: the philosophy of knowledge, epistemology, and the philosophy of understanding and explanation.

We want to distinguish knowledge from right opinion. This is a distinction from Plato's Meno, amplified by John Stuart Mill's work on freedom of speech (Plato 380AD; Mill 1869). (Nowadays, 'right opinion' would probably be more usually described as being 'true belief'.) Plato defined knowledge as having three components:

> Knowledge needs to be true.
> It needs to be believed.
> It needs to be justified.

An Oracle ML might provide right opinion, but it does not give knowledge. For knowledge, the knower has to have evidence and be able to provide a reasoned defense, or explanation, or justification of the known. In brief, the knower has to be able to give the reasons why. Socrates argued:

> ...true opinions: while they abide with us they are beautiful and fruitful, but they run away out of the human soul, and do not remain long, and therefore they are not of much value until they are fastened by the tie of the cause...when they are bound... they have the nature of knowledge; and ... they are abiding. And this is why knowledge is more honourable and excellent than true opinion, because fastened by a chain (Plato 380AD, 98).

An Oracle does not give knowledge. It does not give causes.

Then, what, in this context, is understanding?

> A simple but plausible answer given by contemporary philosophers of science is as follows: to understand a phenomenon is to grasp how the phenomenon is caused (Strevens 2013).

The phenomena of interest here is how the successful ML systems produce the correct predictions that they do. To understand, we need to know the causes at work. With Oracles, the causes are exactly what we do not have. Hence, to address these considerations, there is the existence of the research field Explainable AI (XAI), with its own techniques and methods. Space does not permit extended discussion here of XAI.

There are actually two causal problems with Oracles. There is the problem of specific predictions, understanding why a specific prediction has been made in a particular case, and there is the problem of understanding how a given Oracle works as a whole inside its black box. An example of the first problem is when a radiologist works in partnership with an ML system to assess a patient and to diagnose possible cancer. The radiologist (and the patient) need to know the basis of, or evidence for, the ML system's prediction. The reason for that need is that knowledge is better than right opinion. Knowledge is better for trust, ethics, respect for human beings, decision making, regulatory and legal requirements, etc. An example of the second problem is that the designers or programmers of the radiology image diagnosis system should understand how it works. That would help them going forward to assess and improve the system (Strevens 2013).

Elsewhere, we do negotiate our lives using correlations (right opinions). The farmer puts fertilizer on crops, we take aspirin for headaches, and a majority of us avoid smoking for health reasons. Some of these connections are undoubtedly causal. But we usually do not know of the details of the causality. Often, this is the best we can do. The phenomena, the laws, if

there are any, the initial conditions, and the interactions of these, are all intertwined and complex. Often, they are also probabilistic. Basically, the systems are close to being black boxes. If XAI can do as well as agriculture and medical science, that might be good enough.

13.10 Annotated Readings for Chapter 12

Druga, Stefania, Sarah T. Vu, Eesh Likhith, and Tammy Qiu. "Inclusive AI Literacy for Kids around the World." In Proceedings of FabLearn 2019, 104–11. FL2019. New York, NY, USA: Association for Computing Machinery, 2019. https://doi.org/10.1145/3311890.3311904. (Druga et al. 2019b)

Gesser, Avi, Robert Maddox, Anna Gressel, Mengyi Xu, Samuel J. Allaman, and Andres S. Gutierrez. "New Automated Decision-Making Laws: Four Tips for Compliance." Debevoise Data Blog (blog), 2022. https://www.debevoisedatablog.com/2022/06/25/new-automated-decision-making-laws-four-tips-for-compliance/. (Gesser et al. 2022)

Goodman, Bryce, and Seth Flaxman. "European Union Regulations on Algorithmic Decision-Making and a 'Right to Explanation.'" AI Magazine 38, no. 3 (2017): 50–57. https://doi.org/10.1609/aimag.v38i3.2741. (Goodman and Flaxman 2017)

Mittelsteadt, Matthew G. "Artificial Intelligence: An Introduction for Policymakers | Mercatus Center," 2023. https://www.mercatus.org/research/research-papers/artificial-intelligence-intro-for-policymakers. (Mittelsteadt 2023)

Nori, Harsha, Samuel Jenkins, Paul Koch, and Rich Caruana. "InterpretML." C++. 2019. Reprint, InterpretML, 2023. https://github.com/interpretml/interpret. (Nori et al. [2019] 2023)

Pritchard, Duncan, John Turri, and J. Adam Carter. "The Value of Knowledge." In The Stanford Encyclopedia of Philosophy, edited by Edward N. Zalta and Uri Nodelman, Fall 2022. Metaphysics Research Lab, Stanford University, 2022. (Pritchard, Turri, and Carter 2022) https://plato.stanford.edu/archives/fall2022/entries/knowledge-value/. This is a valuable resource on considerations of knowledge versus right opinion.

Ridley, Michael, and Danica Pawlick-Potts. "Algorithmic Literacy and the Role for Libraries." Information Technology and Libraries 40, no. 2 (2021). https://doi.org/10.6017/ital.v40i2.12963. (Ridley and Pawlick-Potts 2021b)

Stanford HAI. “Generative AI: Perspectives from Stanford HAI,” 2023. https://hai.stanford.edu/sites/default/files/2023-03/Generative_AI_HAI_Perspectives.pdf. (Stanford HAI 2023b) These are perspectives from Stanford leaders in medicine, science, engineering, humanities, and the social sciences on how generative AI might affect their fields and our world.

Sulmont, Elisabeth, Elizabeth Patitsas, and Jeremy R. Cooperstock. “Can You Teach Me To Machine Learn?” In Proceedings of the 50th ACM Technical Symposium on Computer Science Education, 948–54. SIGCSE ’19. New York, NY, USA: Association for Computing Machinery, 2019. https://doi.org/10.1145/3287324.3287392. (Sulmont, Patitsas, and Cooperstock 2019)

Chapter 14: Librarians as Managers

14.1 Coming on Board

Libraries need management, and many librarians have management duties. Of course, they are usually assisted in these duties by computers and general library automation. The use of AI or ML in this setting would be a step further. Amanda Wheatley and Sandy Hervieux's have published an environmental scan of the use of AI in academic libraries (Wheatley and Hervieux 2019).

Their appraisal is bleak. Here are some quotations from the Wheatley and Hervieux paper, offered without comment or judgement:

> What is perhaps the library's best kept secret has been its slow uptake on automation and digital technologies.
>
> [On the possibility of libraries being run by machines] This unsettling reality should have awoken librarians, instead the profession reacted as it did to most technological revolutions - it waited. In fact, it is still waiting.
>
> The absence of scholarly research on AI-related technologies in libraries is not to be unexpected. Libraries have suffered from issues on the adoption of digital technologies and a general resistance to change throughout the twentieth and twenty-first centuries.
>
> The progression of industrial and office automation paved the way for libraries to adopt similar technology, yet this adoption was always years behind the current trends.

> The current state of artificial intelligence in academic libraries has proven to be nearly non-existent.

Separately from this environmental scan, there are literature reviews that range wider than academic libraries. Rajesh Das and Mohammad Islam, in their systematic literature review of the application of AI and ML in libraries, identify in the publications the following ten themes (R. K. Das and Islam 2021):

- collection building and management,
- processing in libraries,
- circulation and user studies,
- reference service,
- library administration,
- library customization and retrieval,
- research and scholarship,
- service quality and innovation,
- intelligent agents for information search and retrieval,
- study on implementation and existing technologies and solution

Their review is backward looking, of necessity. It is considering what has been done (prior to 2020) and what AI and ML techniques were used to do it.

There is also Andrew Cox, Stephen Pinfield, and Sophie Rutter's *The intelligent library: Thought leaders' views on the likely impact of artificial intelligence on academic libraries* (Cox, Pinfield, and Rutter 2019). This

does have valuable content— its editorial summary is excellent. But it is old, and it is not clear how well informed the 33 'thought leaders' were around 2018. One of them did not know what Artificial Intelligence was, and none of them knew about large language models (nor did the rest of us).

IFLA is The International Federation of Library Associations and Institutions. It styles itself as 'the Global Voice of the Library and Information Profession'. Its 2022 Trend Report Update has no mention of artificial intelligence or machine learning (Al Badi et al. 2023). IFLA does have its earlier Statement on Libraries and Artificial Intelligence, which is very good (IFLA 2020).

Barbara Wood and David Evans write in their paper *Librarians' Perceptions of Artificial Intelligence and Its Potential Impact on the Profession*:

> The results of our survey point to an overwhelming sense of complacency among librarians in regard to the transformative/disruptive effects of this technology. For the past 35 years, academic libraries have successfully embraced computerization. Why is it that, at this time, we have our heads in the sand? We liken it to the climate change debate—the data is there, but we choose to ignore it (Wood and Evans 2018).

For us, it is worthwhile for us to consider aspects of these themes which we have not considered elsewhere. In Chapter 10 the following topics were mentioned

- Workflow and Improving Service

- Optimize the Use of Space (and, Indeed, Other Resources)
- Robots
- Mimicking Librarian Experts' Behaviors

We are interested specifically in machine learning and artificial intelligence (not in plain automation). We are using the (R. K. Das and Islam 2021) as the skeleton for this.

14.2 Data and Analyses

There will be data on resources, processes, user behavior, and more. From these, there can be

- **Predictive Analytics** This identifies past usage patterns and trends as input to predict demand and need. It could be looking at many types of data as input (demographics, time of day, time of year (e.g. exams, summer breaks)). With machine learning it does not really matter how rich or apparently irrelevant the types of data. After training, anything known that is not connected with the correlations will just be given weight zero (i.e. discounted).
- **User Behavior Analytics**. This is data on what the patrons are doing, either individually or collectively. Attention needs to be given here to privacy or informed consent. Anonymous data, for example on which books are frequently checked out, can help with decisions on which books to keep physically in the stacks and which to store elsewhere, perhaps off-site. In contrast, recommendation systems will

work better with information tied to patrons individually (that still can be kept anonymous).

- **Learning Analytics**. In cases where libraries are serving students, faculty, and research, there can be data on what resources the patrons use and how they use them for the purposes of teaching and learning. In turn, this can be used to improve the process of education. There is a large and burgeoning field Artificial Intelligence in Education (AIED). See, for example, (Stanford HAI 2023a).) Librarianship will have a role in this. That role will extend to School Libraries and Public Libraries.

14.3 Evidence-Based Librarianship

When making decisions, evidence-based practices encourage the use of the best available evidence, in conjunction with expertise and attention to values (Wikipedia 2023d). This approach tends to downplay tradition, especially when no-one knows why the tradition is as it is. There is Evidence-Based Library and Information Practice (EBLIP), with supporting books, research articles, and a journal (for example, (Hjørland 2011; University of Alberta Library 2023). Maybe now is its time. There is now plenty of data, and ML can analyze a quantity of it in a way that no human can. Of course, the management and administration of libraries used any amount of data prior to the advent of EBLIP. But now there is much more data and the existence of ML that can interact with that data intelligently. As mentioned many times, ML can deal with volume in a way that collective human resources cannot.

14.4 Data-Driven Decision Making

14.4.1 Collection Building and Management

ML can analyze the collection and its uses. Predictive Analytics can be used as part input on decisions on which materials to purchase, keep in the collection, or weed or discard. It can help with resource allocation. The analysis would take input from availability of materials in other libraries, or the needs of the library's users. (See (Litsey and Mauldin 2018).)

14.4.2 Circulation and User Studies

Service, recommendation, personalization, and AIED all have relevance here. ML together with the data analytics mentioned earlier have the potential to effect improvements.

14.4.3 Processing in Libraries

AI can help with many of the processes that are part of librarianship. Most of these have been mentioned earlier in this text, including cataloguing, classification, acquisition, archiving, digitization, transcription, translation, indexing, and summarization.

14.4.4 Research and Scholarship

The view here is that shortly the information discovery techniques and tools are going to be far superior to the ones we have today. We have amplified on this thesis throughout this text. What managers should do is to support the use of these tools.

14.4.5 Service Quality

There will be improved metrics on most aspects of service. Being able to measure how good service is does not of itself guarantee good service. But whenever insights or innovations come to the fore, it is good to know whether they actually do make service better.

14.5 Acquiring the Appropriate AI Tools

The managers can acquire the relevant AI tools.

IFLA makes a suggestion that would kill another bird with that stone:

> Libraries can also support ethical AI research and development by their procurement choices: purchasing AI technologies which abide by ethical standards of privacy and inclusivity. This would both reaffirm the trust of users in libraries, and send a message to the AI research field by increasing the demand for ethical AI technologies (IFLA 2020, 2).

14.6 Analysts and Staff

There is a shortage of skills and awareness.

> The demands of responding to such changes may reveal a significant skills gap in the sector. We know there is already strong demand in the economy for data scientists for data analysis and visualisation. Perhaps some librarians will be required to develop these skills, or at least awareness of different techniques and how they need to be supported. These demands are a challenge because they prompt librarians to learn more about IT and quantitative data analysis, including statistics. In a relatively low-paid sector we may be unlikely to attract people with stronger STM backgrounds to the profession, while those in the profession are typically from an arts background. More optimistically, we can say there will also be a need for librarians as data curators to take on new data (Cox, Pinfield, and Rutter 2019, 17)

Foundation Models have given us all a break here. You do not have to be a programmer or a mathematician to use, develop, or configure LLMs. You can even be from an 'arts background'. So-called 'Prompt Engineering' makes many things possible— we will look further into that in Appendix B.

14.7 Fear of AI

Within the library profession there does seem to be a fear of AI (as evidence, see, for example, (Cox, Pinfield, and Rutter 2019; ExLibris 2019)).

ExLibris invites us librarians to be Liberty Leading the People (Wikipedia 2023h):

> … embracing AI not as users, but as active players to fight the risks of bias, misuse, and discrimination If libraries take an active role in the implementation of artificial intelligence applications in the information management landscape, then they can help programmers find the best data for their algorithms. Once they assume the leading role, librarians can be co-creators of "an intelligent information system that respects the sources, engages critical inquiry, fosters imagination, and supports human learning and knowledge creation," according to Catherine Nicole Coleman. AI solutions can also facilitate both more process transparency and greater data control, with libraries able to safeguard their most important principles and maintain trust, neutrality, freedom of expression, mindful media consumption, and equal access to information, while promoting digital inclusion and data privacy
>
> "Advocacy should not be directed at maintaining traditional librarianship, but in influencing the development of the emerging information systems."
>
> (ExLibris 2019)

14.8 Annotated Readings for Chapter 14

Gale, part of Cengage Group. “Library Marketing: Improve Outreach with Gale Analytics,” 2023. https://www.gale.com/databases/gale-analytics. (Gale, part of Cengage Group 2023)

Litsey, Ryan, and Weston Mauldin. “Knowing What the Patron Wants: Using Predictive Analytics to Transform Library Decision Making.” The Journal of Academic Librarianship 44, no. 1 (2018): 140–44. https://doi.org/10.1016/j.acalib.2017.09.004. (Litsey and Mauldin 2018) This is old now (before LLMs). One commentator describes that it '… seems like a distopian vision that thinks it's a utopian vision'.

Chapter 15: Librarians as Astronauts

15.1 Astronaut Training

Well, you are in a good place. To do creative work with Foundation Models, you do not need to know anything about machine learning, or computer programming. You do not even have to have any domain knowledge about the area you are going to work in (which we are assuming here to be librarianship). The Foundation Model you use will have all the knowledge that is required. Your role is to ask it in ordinary natural language— to 'prompt' it— to do whatever you are looking for. Over the past twenty years or so many librarians have become absolute experts at using search engines, say Google Search. Now there are new kids on the block, Foundation Models, Large Language Models or Large Multimodal Models. It would not be a bad thing at all for many librarians to become absolute experts at LLMs or LMMs.

15.2 Why Should You Learn How To Do It?

- Gives you understanding
- Allows you to teach others
- Allows you to produce apps.
- Allows you to work with ML professionals

15.3 What are the Real Creative Possibilities

This is hard to know (stating the obvious). Three areas where modern machine learning might have a distinction advantage over the incumbents in librarianship or in information curation and provision are: data visualization, chatbots, and information discovery including text data mining.

Data science— making sense of data— is important. Having the right representation of the data is often critical to understanding that data and seeing how to reason with it. The key point was made by Jill Larkin and Herbert Simon in their paper *Why a diagram is (sometimes) worth ten thousand words*. They suggest that informational equivalence should be distinguished from computational equivalence (Larkin and Simon 1987). The distinction can be explained by means of an illustration.

Donald Norman drew attention to a famous example of two games (Norman 1993). In the first:

> you and your opponent take turns in choosing and selecting single, previously untaken, numbers in the range 1..9, and the winner of the game is the first person who has 3 numbers that sum to 15. [So you try both to get numbers that sum to 15 and also to prevent your opponent from getting such numbers.]

In the second:

> you play Tic-Tac-Toe (or Noughts and Crosses),

Now, the first game we would find awkward and might lose from time to time (until we devised a suitable representational scheme). The second is trivial, and an attentive adult would never lose a game. But the two games are the same. They are different representations of the same game. If the numbers are laid out as though they were on a Tic-Tac-Toe board:

6 1 8

7 5 3

2 9 4

Then the problem of finding three numbers that sum to 15 is exactly that of finding three in a vertical, horizontal, or diagonal line.

Any information as to the board's state, or to the numbers selected, is exactly inter-translatable back and forth between Tic-Tac-Toe representation and numbers-selected representation. There is information equivalence. But humans can manipulate and compute with one representation much more easily than the other. (And, interestingly enough, a computer, or a computer program, would be more at home with the numeric representation as opposed to the two-dimensional board representation.)

Information opens up different possibilities for an agent depending on how it is represented. This is matter of the manipulations, operations, and computations, that the depiction facilitates. (A good collection of historical depictions of statistical data can be found in Michael Friendly's *Gallery of*

Data Visualization (Friendly 2007).) Machine learning may well be able to learn the best ways to represent data.

Let us sketch some territories here. In so far as this concerns diagrams, as it would for education and for research by humans, the area of interest is cognitive psychology and heuristics. In so far as it concerns computer processing, it would be a matter of data-structures and algorithms— parts of computer science. Librarians may be fine with either of these, but they are a little outside librarians' bailiwick.

Chatbots have been discussed extensively in this text. Certainly, chatbots in libraries represent an opportunity, possibly even a research opportunity. But chatbots for the purpose of service are going to be everywhere. It may be hard for librarian developers to keep themselves clear of the development stampede.

Information discovery and text data mining are both gifts to librarians. Librarians know from text. We will amplify shortly on one aspect of this possibility.

15.4 Sitting in Your Tin Can

If you are going to sit in your tin can, you could think about how ML might provide new ways to interact with library resources. You could think about interactive games and simulations that might help with access or with AI literacy. You could think about many things.

15.5 Exploring World 3

15.5.1 Undiscovered Public Knowledge (UPK)

In 1986 Don Swanson wrote as the abstract to his paper *Undiscovered Public Knowledge*:

> Knowledge can be public, yet undiscovered, if independently created fragments are logically related but never retrieved, brought together, and interpreted. Information retrieval, although essential for assembling such fragments, is always problematic. The search process, like a scientific theory, can be criticized and improved, but can never be verified as capable of retrieving all information relevant to a problem or theory. This essential incompleteness of search and retrieval therefore makes possible, and plausible, the existence of undiscovered public knowledge. Three examples intended to throw light on the logic of undiscovered knowledge are constructed and analyzed. The argument is developed within the framework of a Popperian or critical approach within science and on Popper's distinction between subjective and objective knowledge--the distinction between World 2 and World 3. (Swanson 1986)

The referenced work of Karl Popper can be explained thus. Popper had introduced in the idea of three worlds: World 1 (the physical world), World 2 (the mental world), and World 3 ('... the world of objective contents of thought, especially of scientific and poetic thoughts and of works of art.'). World 3— objective knowledge— contains the *contents* of books and libraries, not the physical books (World 1), nor the mental ideas conjured

up by the books (World 2), but the contents of the books understood as the objective assertions they make. (See *Epistemology Without a Knowing Subject* (Popper 1968).) World 3 objects can stand in logical relations one to another. One book may contain the assertion 'Leda is a swan'. Another book may contain the assertion 'All swans are white'. Those two books together, considered as a combined whole, entail the consequence that Leda is white. Swanson instantiated and generalized this in the following way. Suppose (some recorded research tells that) process A causes effect B and (some entirely different recorded research tells that) B causes effect C then A causes C, and this, perhaps hitherto unknown, relationship can be discovered by looking at the recorded research. This is an example of *Undiscovered Public Knowledge* (UPK). A real scientific example that Swanson offered from 1986 is that it was known, as recorded objective knowledge, that (A) fish oil reduces platelet clumping (B) and reduced platelet clumping is (C) of benefit to patients with Raynaud's disease. But it was *not* known as objective knowledge (i.e. published anywhere as recorded knowledge) that (A) fish oil can be (C) of benefit to patients with Raynaud's disease. That relationship can be discovered without entering a laboratory. It can be discovered without leaving a library.

> [Usually] .... different assertions and findings need to be assembled across documents to create a new coherent assertion, much as different pieces of a puzzle are assembled to create a single picture. (Smalheiser 2017, 3)

We should keep in mind here that all knowledge is fallible and conjectural. So that when we appear to discover that fish oil helps with Raynaud's disease, we are discovering a new conjecture, that apparently no one had

proposed earlier, and that conjecture would need testing. If its components had been thoroughly tested and were reliable, it may be that further testing could be minimal. That would depend in part on what the practical consequences would be of trusting the hypothesis. In the case of medical treatments, and drugs, there are established protocols.

It may be that the apparent conjecture from UPK had been known and discovered earlier. Researchers looking for treatments for Raynaud's disease may have had hundreds of conjectures as to what might be suitable, and their problem might have been how to spend their time and money. In this case, the ABC example given above screams loudly: try fish oil. UPK is still valuable.

There are other kinds of UPK (Smalheiser 2017). There can be publications that no one currently reads. These may be publications in truly obscure journals, or in outlets that retrieval systems have trouble in finding (perhaps due to poor indexing or metadata).

Investigators in these areas often talk of one node, or two node, searches or of open, or closed, searches. If the search starts with process A and is trying to find anything it relates to, that would be a one node, or open, search. If the search starts with two processes A and C, and is trying to find a B that connects them, that would be a two node, or closed, search.

[Don Swanson was a librarian. More strictly, he rose to being Dean of the Graduate Library School of the University of Chicago.]

15.5.2 Literature-Based Discovery (Text Based Informatics)

The seminal work of Don Swanson has been generalized and developed to form *Literature-Based Discovery* (sometimes called *Text Based Informatics*). This is used widely in bioinformatics and medical informatics. See, for example, (Crichton et al. 2020; Heatley 2023; Moreau 2023; Sebastian, Siew, and Orimaye 2017; Thilakaratne, Falkner, and Atapattu 2020; Wikipedia 2023j).

A consideration of importance to us is that Literature-Based Discovery until about 2020 used what we now would regard as being primitive bibliometric tools. Computers were used, and Google-style searches. It used authors, titles, keywords, subject headings, etc. But 2023 machine learning, especially LLMs, open up new possibilities. No doubt, UPK researchers are on to LLMs already. But the target domain is vast— every book in every library. There are opportunities for all.

15.5.3 A Message to Librarian Astronauts

> The growth of scientific knowledge is usually thought of in terms of wresting new discoveries from the physical world— World 1— admittedly a world that offers unlimited opportunity for discovery. But it should be of interest to librarians to notice that World 3 also qualifies as an endless frontier and to understand how and why this is so (Swanson 1986, 117)

15.6 Annotated Readings for Chapter 15

Smalheiser, Neil R. “Rediscovering Don Swanson: The Past, Present and Future of Literature-Based Discovery.” *Journal of Data and Information Science (Warsaw, Poland)* 2, no. 4 (2017): 43–64. https://doi.org/10.1515/jdis-2017-0019. (Smalheiser 2017)

For here
Am I sitting in my tin can
Far above the world
Planet Earth is Blue
And there is nothing I can do
(David Bowie. *Space Oddity*)

Appendix A: Some Theoretical Background to Librarianship

A.1 Concepts, Classification, Taxonomies, and Items

A central component of classification is concepts, and a whole collection of concepts, being used at once, amount to a *taxonomy* or *conceptual scheme* or *classification scheme*. A concept has *intension* (meaning) and *extension* (which is the collection, or class, or set, of those items, or individuals, or things, in the world that fall under it, or are instances of it). Concepts will usually have *labels* (i.e. words, or names) that identify them. Some concepts can have two or more different labels— such labels would be synonyms e.g. 'attorney' and 'lawyer'. Some labels can label two different concepts— such labels would be homographs e.g. 'bank' (of river) and 'bank' (financial institution).

There is a general looseness of terminology in classification areas. The terms 'concept', 'class', 'set', 'category', and probably other terms, can get used interchangeably. We will go with the flow on this (for example, often using 'class' or 'set' for concept).

A taxonomy— unless it is a simple list or dictionary— will always have at least one class which is a subclass of another class. Then the overall structure of the scheme might be a 'hierarchy', a 'polyhierarchy', a 'directed acyclic graph', a 'tree', a 'forest of trees', or some other graph-theoretic structure (see (Frické 2012)).

A.2 Controlled Vocabularies, and Thesauri

In many settings, librarianship will use *Controlled Vocabularies* (CVs). A controlled vocabulary is collection of words, or 'terms', where the words are 'controlled'— they have a fixed and definite form. As to the value of CVs, Elaine Svenonius writes

> Perhaps as near as one can come to generalization about the value of a CV is simply to say where precision and recall are important retrieval objectives, then a CV of some kind is mandated. (Svenonius 2003, 837)

[There are more detailed explanations of CVs, and their value, in (Frické 2012; Harpring 2020; ANSI/NISO 2010; Zeng 2005). CVs are still of value in the age of computers and machine learning as they provide a standard way to describe concepts and their relationships, and a standard interface between different systems.]

The simplest CVs amount to little more than lists of terms. CVs used for indexing and metadata (e.g. subject classification) tend to be more sophisticated than this. They have a structure in the background, with links or connections between some terms. As an example, the Library of Congress Subjects Headings (LCSH) is a controlled vocabulary. LCSH has hundreds of thousands of terms in it, and it took a hundred years or more to build. It is probably the biggest and most elaborate CV ever made. Its size and maturity are not the only points of interest. It also relates broader

and narrower terms, and it suggests preferred terms in the case of synonyms.

Indexes typically use CVs. Each hierarchy in an index-CV is a 'tree'. But the index as a whole *is not a single tree*. There is not one single term at the top with every other term being a subterm of it. Rather there are several different trees. The structure is that of a *forest*. For example, the MESH subject headings are in 16 different trees. As an illustration, here are six of them

- Anatomy
- Organisms
- Diseases
- Chemicals and Drugs
- Analytical, Diagnostic and Therapeutic Techniques, and Equipment
- Psychiatry and Psychology

Thesauri are controlled vocabularies with relations provided between broader and narrower words or terms. However, some extant controlled vocabularies themselves have broader and narrow terms (e.g. the aforementioned LCSH). If this practice is acceptable, Thesauri can be much the same as Controlled Vocabularies.

A few words are in order about the nature of the links or relations between the broader and narrower terms. International standards for thesauri distinguish three possibilities here: subclass, instance, and part (ANSI/NISO 2010). Whales are a subclass of mammals, Moby Dick is an instance of a whale, a blowhole is part of a whale. So, an imaginary CV

might contain: 'mammal' narrower term 'whale', 'whale' narrower term 'Moby Dick, and 'whale' narrower term 'blowhole (whale)'. The first relation would be subclass (sometimes called 'generic'), the second instance, and the third partitive. The first two relations can come under the general approach of taxonomies. The third, addressing parts, is usually considered to be mereology (which is the study of parts and wholes).

We will usually assume that the collection of top-level terms in a thesaurus (those that are not narrower terms of other broader terms) could be printed, displayed, or written out in 'alphabetical' order. There is a small wrinkle here. Just what 'alphabetical' order amounts to is a question. To give an example, for a telephone directory, should 'Newark' come before 'New York, or after it? where should numbers go? where should foreign words go? where should abbreviations go? The practice here is known as the 'filing order', and it is a matter of several decisions or conventions (which we will not go into).

A.3 Ontologies and Ontological Vocabularies

An ontology identifies what exists in a certain domain or area i.e. the individuals, the properties, the classes, etc. As examples, a medical setting might have an ontology with doctors, patients, medicines, bacteria, etc. in it. A soccer game might have players, teams, balls, referees, pitches, and the like. A library might have patrons, librarians, books, desks, reading rooms, and similar items. Computer scientists use ontologies as a guide to the requisite data-structures needed for their programs. Librarianship is not really interested in ontologies as such, but it is interested in the words or

terms that are used. An ontology will have terms or words associated with the items that appear. So, for example, in a soccer ontology, 'pitch', 'field', and 'paddock' are near synonyms for the surface on which the game is played. Such terms could be built into a controlled vocabulary, with 'pitch' being the preferred term, and 'field' and 'paddock' being lead-in terms. A thesaurus could also easily be used here (e.g. with 'soccer complex' having 'pitch' as a narrower term). Ontologies themselves, or what we might call 'Ontological Vocabularies', could also be used. For example, libraries lend books (or, perhaps, 'libraries issue books'); we can make one of these forms a preferred form and record in the Ontological Vocabulary that libraries lend books.

Thus, Ontological Vocabularies, in so far as they provide relationships between words or terms, are a more general case than Thesauri. For example, consider, in a medical setting, the relationships '... is a symptom of ...' , '... may be treated by ...' ; a relevant Ontological Vocabulary might have preferred terms for 'headache', 'fever', 'influenza', and 'aspirin', and also such relationships as 'headache may be treated by aspirin'.

If these possibilities are thought of in a graph-theoretic sense: a thesaurus links narrower terms to broader terms, and often the structure of this will amount to a tree i.e. there is a root, and parent nodes can have many children but each child can have only one parent. Some thesauri, for example the LCSH, permit some terms to have more than one broader term (i.e. a child node might have more than one parent node) in which case the structure would be a directed acyclic graph (DAG). In an *Ontological Vocabulary*, any node might possibly be linked to any other node; this

would amount to a general graph (i.e. a web of nodes and links). Roughly, then, an ontology is a generalization of a taxonomy, and an ontological vocabulary is a generalization of a thesaurus.

A.4 Objective, Intersubjective, and Subjective

These three terms are selectors among knowledge, or among candidates for knowledge. When an Art Historian describes a painting as being in the shape or proportions of a golden rectangle, she is picking out a property of the painting. She is making a perfectly objective ascription about the external world. She can be wrong, of course. There can be some mistake with her eyes or means of measurement. Nevertheless, right or wrong, she is aiming to be objective. Moving on, suppose her aesthetic is that of Neoclassicism, to the exclusion of all else. She may describe a painting by Jacques-Louis David as being a good painting. That would be an intersubjective judgement. There are plenty of other Neoclassicists and they would have the same view. But Neoclassicism itself is not part of the external world, it is an aspect of a certain kind of culture. Finally, there is the rest of us who know nothing about art but know what we like. Expressions of what we like would be subjective.

Taxonomies also can be objective, intersubjective, or subjective. Objective schemes are the province of science and mathematics. Classifying matter into atoms, elements, compounds, and mixtures, or classifying the numbers into integers, reals, or rational numbers, or classifying parts of the biological world into genus and species are all objective schemes. As noted earlier, our knowledge of the objective is fallible— scientists or

mathematicians can be mistaken (and hopefully they can correct the errors). Intersubjective schemes arise in the context of society, culture, politics, religion, myths, and the law. A good example is the **Nursing Interventions Classification (NIC)**

> **[it]** is a comprehensive, research-based, standardized classification of interventions that nurses perform. It is useful for clinical documentation, communication of care across settings, integration of data across systems and settings, effectiveness research, productivity measurement, competency evaluation, reimbursement, and curricular design….
>
> **The 565 interventions** in NIC (7th ed.) are grouped into thirty classes and seven domains for ease of use. The 7 domains are: Physiological: Basic, Physiological: Complex, Behavioral, Safety, Family, Health System, and Community. Each intervention has a unique number (code). The classification is continually updated with an ongoing process for feedback and review. In the back of the book, there are instructions for how users can submit suggestions for modifications to existing interventions or propose a new intervention (Butcher et al. 2018).

This is not a description of the external world. This is not intended to tell us how the world is built (the continual updating is a bit a giveaway on this). It is not objective (in our sense of objective). Rather it is a convenient fiction (or, as some might say, it is a social construction). However, nurses, doctors, insurance companies etc. all respect and use this fiction. It is intersubjective. It is artifact they all agree on and use. As to the final category of pure subjective classification— that, for example, is how you and I separately organize our clothes in our wardrobes.

Most classification schemes in information organization are inter-subjective, but some have components that supervene on objective

classifications. For example, schemes for classification of resources on science or mathematics, e.g. books and research papers, would typically partially follow objective classifications that the scientists or mathematicians use.

A.5 Emotive and Descriptive Content

Going back to the Art Historian describing a painting as being in the shape or proportions of a golden rectangle… Being golden rectangular as a property is not something that is either good or bad, and it is not something that she wants to recommend to you or to warn you away from. It is just a plain description. 'golden rectangular' has descriptive content and little or no emotive content. When a viewer describes a painting as being 'nice' or 'good', that viewer is perhaps expressing approval or recommendation. But it is unclear what in particular, if anything, in the painting, 'nice' or 'good' describes. 'Nice' or 'good' have emotive content and little or no fixed descriptive content. (Emotive content is sometimes called 'connotation'.)

Many adjectives (properties, nouns) have both some descriptive content and some emotive content. And it is possible for different words to have the same descriptive content but different emotive content. For example, there are Bertrand Russell's well-known 'emotive conjugations' [Comment, *Brains Trust* (BBC Radio) (26 Apr 1948)] such as

> I am firm; you are obstinate; he is a pig-headed fool.
> I am righteously indignant; you are annoyed; he is making a fuss over nothing.

> We made a tactical withdrawal; our allies retreated; the enemy was routed.
> I have reconsidered; you have changed your mind; he has gone back on his word.

Were the present author to be described as 'firm', or as 'obstinate', or, indeed, as 'a pig-headed fool', the descriptive or cognitive content of those ascriptions is exactly the same (we may suppose). That is the dispositions, and actual actions, of his that these concepts or labels pick out, if correctly used, are identical for all three labels. However, the labels are intended to either express or evoke different sentiments on the part of the speaker or the listener. Roughly, at one end, a speaker that uses 'firm' is mildly admiring of the trait and wishes the listener to be mildly admiring also; at the other end, well, we all know what labeling the author 'a pig-headed fool' is intended to achieve. These expressive or evocative functions are the emotive content. So, each of the concepts [firm, obstinate, pig-headed] has pretty well the same descriptive content, the same intensions, but they differ in emotive content. They are synonyms in all but emotional force.

Of course, both descriptive content and emotive content of particular labels or concepts can change through time and be different across cultures.

For various reasons, usually obvious ones, individuals, groups, and institutions often try to employ different labels or different classification schemes in order to manipulate the emotive-descriptive divide in a way they wish. For example, it may be that the label 'garbage collector' is a term of mild opprobrium, and all concerned might switch to 'sanitary engineer' as the concept or label of choice. Here, the intensions of 'garbage collector'

and 'sanitary engineer' are pretty well the same, but one has unwanted emotional association. [This move is usually a temporary palliative. Say folk do have bad feelings to, or opinions of, the profession of garbage collection— those bad attitudes do not attach to the name rather they attach to what those garbage collectors are and to what they do. We are all familiar with Romeo and Juliet's 'What's in a name? that which we call a rose By any other name would smell as sweet;'. Using the label 'sanitary engineer' might work for a while, but give it 50 years and a person may be able to insult another with the words 'sanitary engineer'.]

A.6 Classification Schemes and the Act of Classification

There are two balls that we are juggling here. There are the classification schemes, and there is the act of classification. Consider, for example, the aforementioned Nursing Intervention Classification (Butcher et al. 2018). It is a perfectly fine classification, absolutely clear as to what it is and what it is supposed to do. Then there is the act of classification i.e. deciding what interventions a nurse, or healthcare provider have done on a particular occasion. As an example, the entry for activities relating to Hypertension includes 50-100 activities. Here are a brief few of those:

> • Instruct at-risk patients to have regular preventative health screenings, including electrocardiogram, echocardiogram, electrolytes, urinalysis, as indicated
> • Instruct related to healthy dietary pattern
> • Instruct related to proper physical activity (e.g., exercise 30 to 45 minutes a day)
> • Instruct related to contributing lifestyle habits that should be avoided (e.g., use of tobacco in any form and alcohol)

• Instruct the patient on lifestyle modification related to sleep and rest patterns (e.g., 8 hours per night is recommended)
• Provide information on possible changes in lifestyle necessary to avoid future complications and control the disease process
• Provide information related to the purpose and benefit of the lifestyle changes
• Instruct related to self-blood pressure monitoring and to report abnormal findings
• Instruct the patient on possible causes of hypertension
• Instruct the patient and family to take an active role in the management of disease process, (e.g., medication indications and administration, maintaining proper diet, exercise and healthy habits, quitting smoking, reducing stress, reducing weight, reducing sodium intake, reducing alcohol consumption, increasing exercise, as indicated)
• Instruct the patient and family on medication usage and indications
• Encourage the patient and family to maintain a list of current medications and reconcile routinely at wellness checks, hospital visits, or hospital admissions
• Instruct the patient to recognize and avoid situations that can cause increased BP (e.g., stress or sudden discontinuation of drug treatment)

(Butcher et al. 2018) p.213-214

Suppose a nurse interacts with a hypertension patient and there is a serious back-and-forth medical conversation. A question is: which of these activities has the nurse done? No doubt the nurse himself, and the local general medical environment know reasonably well. The point being made here is that the act of classification, deciding which items belong in which classes, is fallible, and it is subject to mistakes and possible even to deliberate misuse or malfeasance.

A.7 Annotated Readings for Appendix A

Frické, Martin. *Logic and the Organization of Information*. New York: Springer, 2012. (Frické 2012). This has material on organizational structures and graph theory, and further references that are relevant.

Appendix B: Working With LLMs

B.1 Introduction

Do yourself a favor. Learn as much as you can about LLMs. You can go a long way merely by getting access and by typing. You can go further by having the computer programming language Python on your computer and using an LLM development environment like LangChain.

Many of the major LLM implementations are readily available. As examples:

- Bard (using LaMDA) is available from https://bard.google.com. It is free. Bard implementations will usually have a microphone icon and can take spoken, or dictated, input. [That will make interactions quicker.]
- GPT-4 and ChatGPT are available from https://openai.com. Some modes of access are free. Others can paid for either by usage or by monthly subscription. Current subscription to GPT-Plus is $20 a month. With a subscription, the page https://chat.openai.com/ access to GPT-3.5 and GPT-4, and to plugins and web browsing enhancements. There are about 120 plugins available at the moment (with many more to come, no doubt). You are allowed to have 3 plugins enabled at one time. The author has Wolfram (which gives mathematics and more reliable facts and factual statistics), ScholarAI

(which helps with scientific papers), and Prompt Perfect. Separately the author has an API (Application Programming Interface) key to GPT APIs. This is for programs using a GPT LLM and is charged by the access (usually just cents for a day's work). More on the programming later.

- ChatGPT is also available as a free app for the iPhone (and there will be a free app for Android phones).
- Bing, a search engine (using GPT-4) is available on a web-browser, as an app for smartphones, as part of a chatbot voice assistant on Amazon Echo, and Google Home. New Bing is available from https://www.bing.com/new

We are going to use Application Programming Interfaces (API) for interacting with the LLMs. A valuable resource, with many links to materials is

OpenAI Cookbook (OpenAI [2022] 2023)

B.2 Prompts and Prompt Engineering

An LLM takes some text (or images, or sounds) as an input prompt and returns text (or images, or sounds) as its response. For our purposes, we will generally assume that prompts are given as text in English. Crafting a prompt to obtain the right or desired kind of response is *Prompt Engineering*. Prompt engineering is an active research area, with hundreds of publications. There are also many excellent educational sources for learning prompt engineering. One example is *The Prompt Engineering*

Guide (Saravia 2023) created by Elvis Saravia of Dair.ai (Democratizing Artificial Intelligence Research, Education, and Technologies). That work is clear and comprehensive. We will sketch or paraphrase some of it to give you, the reader, some idea of the strengths and weaknesses of prompt engineering.

A point to be made is that you never really know what quite what you are going to get as the output from an LLM using a prompt. For a start, the relationship between the prompt and the response is probabilistic and usually exactly the same prompt can produce two or three different responses. (If the LLM produces these automatically, they are sometimes called drafts.) Then there is the fact that the textual prompts are given in natural language, say English, and any natural language has a penumbra of uncertainty and ambiguity about meaning. In computer programming, the programs are usually written in a programming language, say Haskell, which has a formal definitive syntax and semantics. There is no doubt whether an expression in the language is well-formed and no doubt as to what it means. In contrast, the LLMs are being prompted by textual strings which may or may not be correct English and which may or may not have an unambiguous meaning. Then the response, the output, is also in possibly ambiguous English in an intended or unintended format. These considerations suggest that there is a trial-and-error, and experimental, aspect to prompting and prompt engineering. In some contexts, this might not matter, when writing fiction for example. But in other contexts, for example, when generating coherent and logically sound legal arguments it might be important.

B.2.1 Basic Examples of Zero-Shot Prompting

Zero-Shot prompting is where you just give the instructions in text (you do not give examples or 'shots').

> '... no demonstrations are allowed and only an instruction in natural language is given to the model' (Brown et al. 2020, 5).

Here are some examples of the following typical NLP processes of interest to us:

- Explanation
- Text Summarization
- Information Extraction, Question Answering
- Text Classification
- Conversation
- Code Generation
- Reasoning

Explanation

> [The prompt]
> Explain prompting used in Large Language Models
> [GPT-3.5 replies]
> Prompting is a technique used in large language models like GPT-3 to generate desired responses or outputs. When using a language model, you typically provide it with a prompt, which is a short piece of text that frames the context or question for the model. The model then generates a continuation of the prompt based on its training and the patterns it has learned from the data.
> Prompting serves multiple purposes:
>
> > Context setting: The prompt helps establish the context for the model, giving it an understanding of the desired topic or

domain. For example, if you want to generate a news article about a specific event, you can provide a prompt that summarizes the key details of the event.

Instructional guidance: The prompt can include explicit instructions or guidelines for the model to follow. These instructions help steer the model's response in a particular direction. For instance, if you want the model to write a poem about nature, you can include instructions like "Write a poem about the beauty of nature and the changing seasons."

Question-answering: You can use prompts to ask specific questions to the model. By providing a question as a prompt, you can expect the model to generate an answer or response based on its understanding of the question and the knowledge it has acquired.

Desired output control: By carefully designing the prompt, you can influence the style, tone, or structure of the generated text. For example, if you want the model to produce a response in a formal or casual tone, you can explicitly specify that in the prompt.

It's important to note that the effectiveness of prompting can vary depending on the language model and the specific task at hand. Experimentation and iteration with different prompts are often required to achieve the desired results.

Text Summarization

Summarize your last response

[GPT-3.5]

Prompting is a technique used in large language models like GPT-3 to guide their output. By providing a prompt, which is a short piece of text, you can establish context, give instructions, ask questions, or control the desired output. Prompts help the model generate more relevant and specific responses based on its training and learned patterns. It's important to experiment and iterate with different prompts to achieve the desired results.

Information Extraction, Question Answering

In your last response you mentioned an LLM, which LLM was that?
[GPT-3.5]
Apologies for the confusion. In my previous response, I mentioned GPT-3 as an example of a large language model (LLM). GPT-3 stands for "Generative Pre-trained Transformer 3" and is one of the most well-known and widely used LLMs developed by OpenAI. It has been trained on a vast amount of text data and is capable of generating coherent and contextually relevant responses based on provided prompts.

What is a good way to learn how to use a LLM? Please keep your answer brief.
[Bard, with the option to view other drafts]
Here are some good ways to learn how to use a LLM:

- Read the documentation. Most LLMs come with extensive documentation that explains how to use them. This is a great place to start learning the basics of how to interact with the model.
- Take a tutorial. There are many tutorials available online that can teach you how to use a LLM. These tutorials can be a great way to learn the basics of how to use the model and how to apply it to different tasks.
- Experiment. The best way to learn how to use a LLM is to experiment with it. Try different prompts and see what kind of responses you get. You can also try using the model to complete different tasks.
- Join a community. There are many online communities where people discuss LLMs. These communities can be a great resource for learning new things about LLMs and getting help from other users.

Here are some additional tips for learning how to use a LLM:

- Start with a simple task. When you're first starting out, it's best to start with a simple task. This will help you learn the basics of how to interact with the model without getting overwhelmed.
- Be patient. LLMs can be slow, especially when they're first being trained. Be patient and give the model time to generate a response.

- Don't be afraid to ask for help. If you're stuck, don't be afraid to ask for help from other users or from the documentation.

With a little practice, you'll be using LLMs like a pro in no time!

Text Classification

Is the following text friendly or confrontational? Text: Hey you get offa my cloud
[Bard]
The text "Hey you get offa my cloud" is confrontational. It uses the words "hey" and "get offa," which are both informal and aggressive. The sentence also contains a direct command, which can be seen as rude.
Here are some ways to make the text more friendly:

- "Excuse me, but could you please get off my cloud?"
- "I'm sorry, but I need to use my cloud. Could you please move?"
- "Would you mind getting off my cloud? I need to use it."

These sentences are all more polite and less confrontational than the original text. They also use more formal language, which can make the request seem more reasonable.

Conversation

Hello, how are you?
[GPT-4 No plugins]
Hello! As an artificial intelligence, I don't have feelings, but I'm here and ready to help you. How can I assist you today?
Would you like to hear how I am?
[ChatGPT-4]
Of course! I'm here to listen. Please feel free to share how you're doing.

Code Generation

Two examples:

Write code to add up some numbers and return their average.
[ChatGPT-4]

Sure, here's a simple Python function to calculate the average of a list of numbers:

```
def average(numbers):
    return sum(numbers) / len(numbers)
numbers = [1, 2, 3, 4, 5]
print(average(numbers))
```

In this code:

- The **average** function takes a list of numbers as an argument.
- It uses the built-in **sum** function to add up the numbers, and the **len** function to find how many numbers there are.
- It returns the sum divided by the count, which is the average.
- The list **numbers** is defined as an example, and the average of its values is printed.

Please replace the numbers in the list with the numbers you'd like to calculate the average for.

write a parser to parse LISP expressions
[GPT-4]
Sure, here's a very basic recursive LISP parser written in Python. This parser will be able to handle simple LISP expressions like **(add 1 2)** or **(mul (add 1 2) (add 3 4))**. Please note that this is a basic implementation and does not cover all aspects of LISP. It's a simplified version to give you a head start.

```
def parse_lisp(expression):
    def tokenize(expression):
        return    expression.replace('(',    '    (
').replace(')', ' ) ').split()

    def read_from_tokens(tokens):
        if len(tokens) == 0:
            raise SyntaxError('unexpected EOF')

        token = tokens.pop(0)
        if token == '(':
            L = []
            while tokens[0] != ')':
                L.append(read_from_tokens(tokens))
```

```
            tokens.pop(0) # pop off ')'
            return L
        elif token == ')':
            raise SyntaxError('unexpected )')
        else:
            try: return int(token)
            except ValueError:
                try: return float(token)
                except      ValueError:      return
str(token)

    tokens = tokenize(expression)
    return read_from_tokens(tokens)
# Example usage:
print(parse_lisp('(add 1 (mul 2 3))'))
```

This code takes a LISP expression as input, tokenizes it (splits it into separate parts), and then recursively parses the tokens into nested lists. Symbols are kept as strings, and numbers are converted to integers or floats.
Remember that this parser won't actually *evaluate* the LISP expressions—it just parses them. If you want to evaluate them as well, you'll need to implement an evaluator that understands the different LISP functions (like **add**, **mul**, etc.).

/*Note from MF. This is not an off-the-chart difficult challenge, but it is not trivial either. The answer is pretty impressive. */

Reasoning

The detective reasoned. If there is blood on the knife, the butler committed the crime. There is no blood on the knife. Therefore the butler did not commit the crime. Is this valid reasoning?
[GPT-4]
The reasoning here seems to follow the format of Modus Tollens, which is a valid form of argument in classical logic. However, the

initial conditional ("If there is blood on the knife, the butler committed the crime.") may not be necessarily true in all cases.
Modus Tollens reasoning goes like this:

1. If P, then Q.
2. Not Q.
3. Therefore, not P.

In this case:

1. If there is blood on the knife (P), then the butler committed the crime (Q).
2. There is no blood on the knife (Not P).
3. Therefore, the butler did not commit the crime (Not Q).

While it is valid reasoning according to Modus Tollens, it is contingent on the truth of the initial conditional statement. In this case, the conditional statement seems to suggest that only the butler could have left blood on the knife, which is quite an assumption. The blood could have come from someone else or the butler might have cleaned the knife, for instance. Therefore, while the logic is structurally sound, the premises (the assumptions or facts that the argument is based on) may not be valid, and thus the conclusion could potentially be flawed.

/*Note from MF. ***GPT-4's answer here basically is completely wrong.*** It pulls up Modus Tollens, which is fine. But then it muddles up its Ps and Qs. Look at the lines numbered 2. The Modus Tollens schema requires Not Q but then it uses Not P as an instance of that.
GPT-4 is attempting reasoning, but does not succeed in this case. */

B.2.2 Examples of Few-Shot Prompting

Few-shot prompting covers cases where examples are given. Outside of LLMs it is very common when giving instructions to give at least one example of what is required (say, when teaching mathematics or passive voice construction in English writing). Most LLMs can work just from examples— they do not need the instruction or explanation.

Here is an example of one-shot. The LLM has been given a single example of an English to French translation and, without even being told what the task is, it has figured out what to do and has done it.

> sea otter => loutre de mer
> peppermint =>
> [GPT-4]
> The translation for "peppermint" in French is "menthe poivrée".

[While this is an example of one-shot prompting, it is unclear how good it is as a test of the LLM. It is from (Brown et al. 2020, 7). That is the main research paper on few-shot prompting. The LLM may have seen it thousands of times during training.]

Here is an example of many-shot.

> cat=>cats
> rabbit=>rabbits
> mouse=>
> [GPT-4]
> The plural form of "mouse" is "mice".

You can see here how problems might arise. The LLM needs to spot the pattern. The two examples can be construed either as forming the plural form or as adding the letter 's' on the end of a word or noun.

> cat=>cats
> rabbit=>rabbits
> mouse=>mouses
> man=>mans
> woman=>

[GPT-4]
The plural of "woman" is "women".

Then

ctt=>ctts
bbit=>bbits
xouse=>xouses
mnn=>mnns
woman=>
[GPT-4]
Following the same pattern you've established with the other words, the transformation of "woman" would be "womans". However, it's important to note that in standard English, the plural of "woman" is "women".

It is just too clever!

B.2.3 Chain of Thought Prompting

Jeff Dean writes

> One of the broad key challenges in artificial intelligence is to build systems that can perform multi-step reasoning, learning to break down complex problems into smaller tasks and combining solutions to those to address the larger problem. (Dean 2023)

When ChatGPT came out, say November 2022, it was clear the available LLMs were weak in this area. Plainly, they could not do mathematical, logical, and scientific reasoning. It was found that to a degree this could be ameliorated by asking the LLM to show its reasoning, to show the steps it made— i.e. to show its chain of thought (Wei and Zhou 2022; Wei et al. 2023). One of the examples that Jason Wei and Denny Zhou give is this

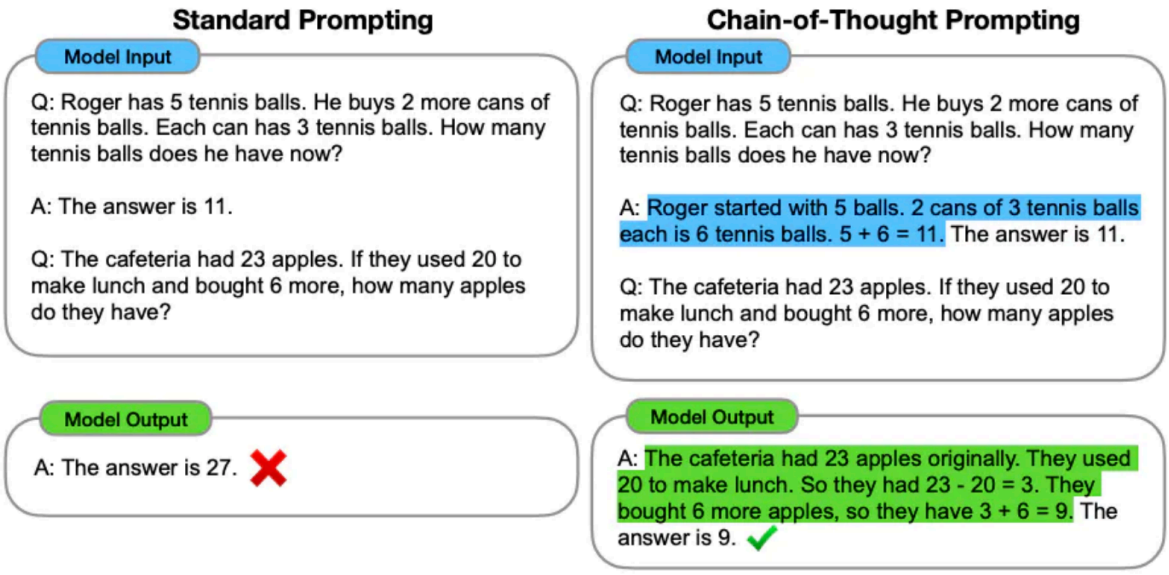


Figure 28. Standard Prompting and Chain-of-Thought Prompting (Wei and Zhou 2022).

There is a one-shot prompt here, showing an example and the example displays the reasoning. It soon became clear that this could be done zero-shot, without examples, merely by adding to the original prompt:

> Let's think step by step.

Then this adjustment of the prompt could be done automatically (for example, by the use of templates, which we will get to). This is an ongoing area of experimentation and discovery. Getting the LLM to think step by step, and to show its reasoning, is a good idea.

B.2.4 Tuning, or Configuring, the Models or Prompts

Most LLMs have a parameter— 'temperature'— that can be set prior to prompting. The background explanation of this is that an LLM's response involves a choice among probabilities. For example, with cloze task like

London is the capital city of [mask]

there will be several possibilities for the blank e.g. 'England', 'culture', 'fashion', 'progress', etc. and these will have different probabilities. Now, if the LLM always choses the most probable response to a word, phrase, sentence, etc., it will always give the same answer. You might not want this. For example, if, on different romantic occasions, like date night once a week, you asked for a love poem for your sweetheart, you presumably would not want the same poem each time. On the other hand, if the LLM always choses improbable answers, the answers may be interesting and humorous, but also often plain wrong. So, there is a parameter 'temperature' which adjusts for this. A higher temperature means that the LLM is more likely to generate creative and interesting answers, while a lower temperature means that LLM is more likely to generate factual and accurate answers. Temperature is in the range 0-1 and it would ordinary be set by default to 0.7. In the case of Bard, it can be set using the following syntax:

bard.generate(prompt, temperature=0.5)

For example:

bard.generate("London is the capital city of ...", temperature=0)

The syntax for setting the temperature in other LLMs may well be different.

We know that typically the large models can make mistakes. They are not 100% reliable. GPT-4V is no different in this regard. We also know that the models can give two contradictory answers to the same prompt in the same context. That such answers are contradictory means that one of these answers is false. That is, more generally, the models can give false answers. Also, in more than a few cases there can be no practical means for the user to check the answer. For example:

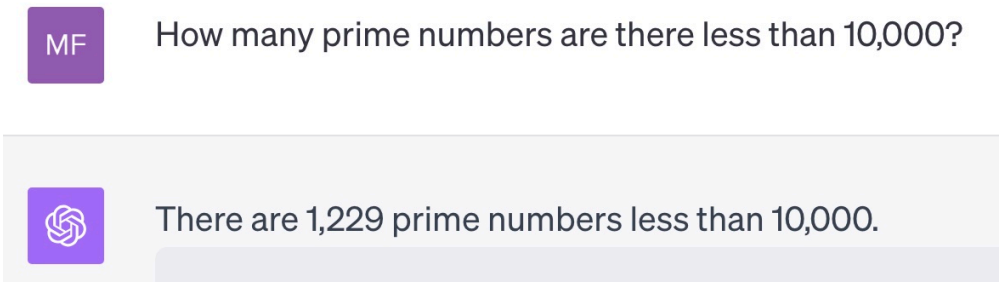


Figure 29. GPT-4V on Prime Numbers.

This answer may be right, or it may be wrong. Typical users would have no idea how to check which it is, or whether they should repose trust in what GPT-4V is telling them.

There are ways to improve the quality of answers. There are two standard techniques: imposing constraints and giving the LMM (or LLM) a reputation.

Constraint prompting is requiring the LMM to answer in a restricted ('constrained') way or format. For example, one constrained prompt to read a driving license would be:

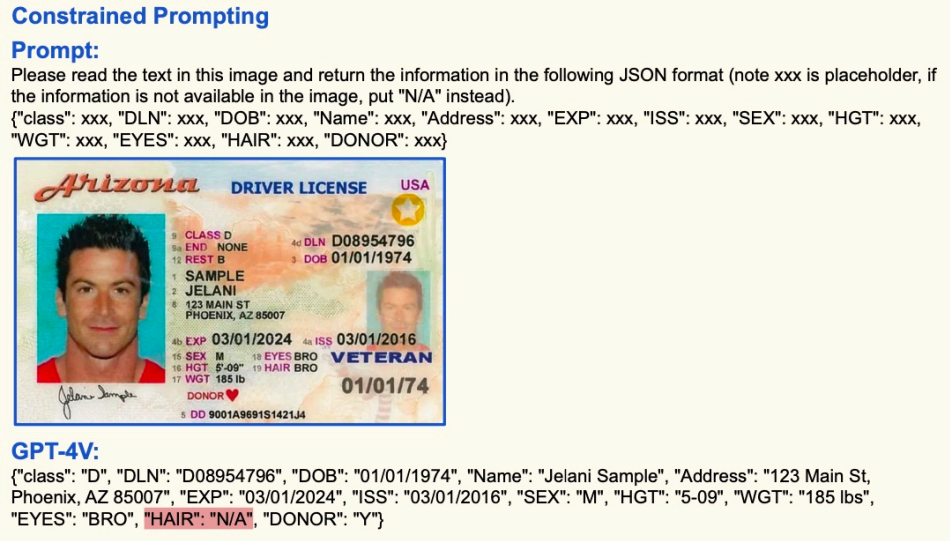


Figure 30. Screenshot of GPT-4V Reading a Driving License from (Yang et al. 2023).

This is better than just prompting the LMM to 'read the text in the image. [Notice that it has made a mistake with the HAIR.]

As explained in Sections 3.10 and 4.7, Base Models of LMMs are trying only to produce the next word as output. They need further training to become Instructor Tuned, or Fine Tuned, LLMs. But even at this point, they are

only doing an extension of their training. Many kinds of instruction or prompting can invoke an activity without necessarily emphasizing that the outcome be correct or true. Yang et al. mention in their paper the prompt 'Count the number of apples in the image', where the image is of apples laid out in rows and columns (Yang et al. 2023). Now, depending on its training, an LMM can count without really focusing on producing a correct answer. A better style of prompt **conditions on good performance**. An example prompt for the apple image would be 'You are an expert in counting things in images. Let us count the number of apples in this image by <and then give detailed instructions on how to do it, and how to check the answer.>' Some of the recent LLMs accept both 'System Prompts' and (Other) 'Prompts'. A system prompt could then be used to set the system as being in the context where it was an expert in counting objects in images. This use of system prompts is often used to enhance 'steerability'. If you would like to 'steer' an LLM to be an be an expert in counting or to answer in the style of Emily Dickinson, you would use a system prompt to do it.

B.3 Choices on Development

Most ML programming work will either use Python or allow Python to be used. In turn Python is generally available on Macs, Windows, and Linux machines. Python will use 'pip' which is a package installer. Pip will usually be installed automatically with Python.

To check whether you have what is needed use a terminal and try the following 3 commands:

```
% python3 --version
Python 3.11.3

% where python3
/Library/Frameworks/Python.framework/Versions/3.11/bin/pyth
on3
/Library/Frameworks/Python.framework/Versions/3.8/bin/pytho
n3
/usr/local/bin/python3
/usr/bin/python3
% pip --version
pip 23.1.2 from
/Library/Frameworks/Python.framework/Versions/3.11/lib/pyth
on3.11/site-packages/pip (python 3.11)
```

The response to you on your machine should be somewhat similar, although probably not exactly the same. If you seem not to have Python, go to https://www.python.org/ and install it.

To give you a general idea where we are going here. We are not going to create any LLMs (which would take months and might cost tens of millions of dollars). We are going to work with existing LLMs (e.g. GPT-4) using their APIs. We will send some messages and get replies, both of which we can process as we would like. There will be short Python programs— usually around 50 lines— to do the work. Typically, these Python programs will have 'import' statements at the top, for example:

```
import langchain
from langchain.llms import OpenAI
```

```
from langchain.utilities import WikipediaAPIWrapper

### langchain this is a convenient host system for working
with LLMs
### OpenAI gives us the API to GPT-3 GPT-4 etc.
### WikipediaAPIWrapper allows us to ask for material from
Wikipedia
```

Of course, starting out you will not have the OpenAI (and other) packages on your computer. This is where the Python package installer pip comes into play. There will be commands like:

```
% pip install -r requirements.txt

% pip install langchain
```

where, the requirements file has a list of, surprise, the requirements. You do not need to worry about this right now.

As to Foundation Models, the LLMs, they can be proprietary or open-sourced. Proprietary are closed-source models, usually vastly expensive to make and are owned by commercial companies. That they are 'closed-source' means that the source computer code is not available to you. You cannot see it, you do not know what it is. This does and does not matter. It does not matter in that there is no possibility of you altering the code and re-pre-training the model, because that might cost $10 million which you do not have to spare. It does matter in that whoever owns the code might change it or withdraw it (they might have a huff, stop playing the game, and take their ball and go home). It also matters in that closed-source may be preferable in that it presumably keeps powerful code out of the hands of

bad actors. It may be possible to get some kind of closed-source version of a Foundation Model, which has locked in parameters. That might protect you from malfeasance, or monopolistic behavior, by the Model's owner. If LLMs develop as it seems they are going to, the models will have a role somewhat similar to public utilities. We need to be cautious when letting private companies have control of public utilities. Some Foundation Models have been open sourced, and are freely available to all (to use, see, or develop). Hugging Face provides a hub, a library of open-source Foundation Models (Hugging Face 2023). When you work with the APIs of OpenAI (or with HuggingFace, or with Pinecone...) you will need to get an API 'key'. Getting these may vary a little from host to host. In the case of OpenAI, go to their website https://openai.com/ , create an account, then View API keys under your profile, and proceed from there. There are charges. Typically, they will give you a $5 credit. The charges are reasonable. There is variation here but, ball-park figures, you can use their API all day for less than a dollar.

Educators working with Python might use either or both of Jupyter notebooks or an Integrated Development Environment (IDE). We should probably follow along with both here.

B.4 Moving Forward With LangChain

B.4.0 A Note on the Status of LangChain and Similar as of 11/6/2023

As mentioned earlier, on 11/6/2023, OpenAI released its builder framework for GPTs and these are assistants, some close to being agents, based on GPT software. Also, many of the techniques that third-party developers had produced in the previous six months to enhance the original ChatGPT (such as being able to process private libraries, say of pdfs) have been rolled in to GPT-4 Turbo. This means that developers, and possibly even some start-up companies, are on shaky ground. OpenAI may have and release better software, or they may simply absorb or mimic any outside software that appears.

In the case of LangChain, it can work with LLMs from many different companies (e.g. from Google, from Meta, etc. — in fact with most of the open-source components from Hugging Face). There is a pedagogic advantage to this— there is no lock-in to the one company (no matter how benevolent that company might be).

The whole AI-LLM landscape is changing rapidly, and no one know exactly what is best. Our view is: doing some work with LangChain is good for learning, but it might be prudent not to form your start-up around LangChain and seek venture capital funding on that basis.

B.4.1 What is LangChain?

> LangChain is a software development framework designed to simplify the creation of applications using large language models (LLMs). As a language model integration framework, LangChain's use-cases largely overlap with those of language models in general, including document analysis and summarization, chatbots, and code analysis. (Wikipedia 2023g)

LangChain was created by Harrison Chase in October 2022 as an open-source project (Chase [2022] 2022). As of May 2023, 836 developers had contributed to it. What it does is:

1. Allows access to LLMs such as closed-source ones from OpenAI (e.g. GPT-4) or open-source ones from Hugging Face (e.g. LLaMA by Meta AI or Flan-T5 by Google).
2. Allows access to software to extract and manipulate text from resources such as pdfs (e.g. a User's research papers, a library's holdings) or other sources (e.g. Wikipedia)
3. Allows access to software that creates embeddings.
4. Allows access to vector databases to store, augment, or retrieve embeddings (such as Pinecone in the cloud) or (FAISS (Facebook AI Similarity Search) on a local machines).
5. Allows the combination of all these into 'chains' which can carry out any or all of the standard NLP, and other, operations (as a really simple example: Find the population density of the capital of Great Britain [This might involve determining the capital of Great Britain; then, having found that, determining its population, determining its area, and dividing one figure by the other].)
6. Allows the combination of chains into 'agents', intelligent assistants which can carry their operations autonomously.

LangChain can be used to build applications such as chatbots, question answering systems, summarization systems, and computer code writing applications. LangChain looks very promising. LangChain has a website

https://langchain.com/, and a conceptual guide https://docs.langchain.com/docs/ You should scan the conceptual guide.

Prompt templates (once you get it right stick with it)

Most of Meta/Facebook LLMs are open sourced.

B.4.2 LangChain Experiments Displayed to a Web Page

There are many excellent Integrated Development Environments (IDEs) for Python. Two of them are: PyCharm from JetBrains https://www.jetbrains.com/pycharm/ (there is a free version), and Visual Studio Code (VSC) from Microsoft https://code.visualstudio.com/ (and this is free). JetBrains have been very supportive of instructional courses that the author has taught— a tip of the hat to them. But, right now, we will go forward with VSC. There are many YouTube videos on how to work with Python and VSC. Here is one:

> Setup Python Using Visual Studio Code On Mac [Sonar Systems, the author, have similar videos for the other Operating Systems.]

The main thing with VSC is that you need to have the extension for Python installed. There is a separate mini-gotcha. When evaluating a Python file, VSC might ask you which Python interpreter to use. Then, when you try to make a choice, it might say there isn't one installed. At this point, it will

allow you to insert a path to an interpreter. Since you have run the **`% where python3`** command (as shown above) you will have path that you can insert.

To get a web page display we will use Streamlit (https://streamlit.io/) which is a fast and convenient way to produce web applications from Python code (particularly from Python ML prototypes). You will need an OpenAI key (getting one has been described earlier). You will need a running Python: either use a VSC installation, as described, or a set-up of your own. You will need to install some Python packages. Open a terminal in VSC and use pip install:

```
% pip install streamlit langchain openAI wikipedia
```

Shortly, you will create a Python file in VSC, say app.py. Then streamlit will run it (from the VSC terminal):

```
% streamlit run app.py
```

That should get you started with LangChain. Let's experiment away!

App Framework

This code will just open a web page for us. We will put our API key in at this point, for use later. Copy and paste the code into a Python file in VSC, say app.py, and run it by executing **`streamlit run app.py`** from a VSC terminal. Streamlit will keep running. You can stop it by typing Ctrl-c :

```
# % pip install streamlit langchain openAI wikipedia

import streamlit as st
import langchain
import os

os.environ["OPENAI_API_KEY"] = <INSERT YOUR API KEY HERE> #
sorry you can't use mine

#Streamlit display
st.title ('Our Assistant')
input = st.text_input ('Write about')

#Testing we have input
if input :
    st.write('The inputted topic is ', input)
```

The resulting web page should look similar this:

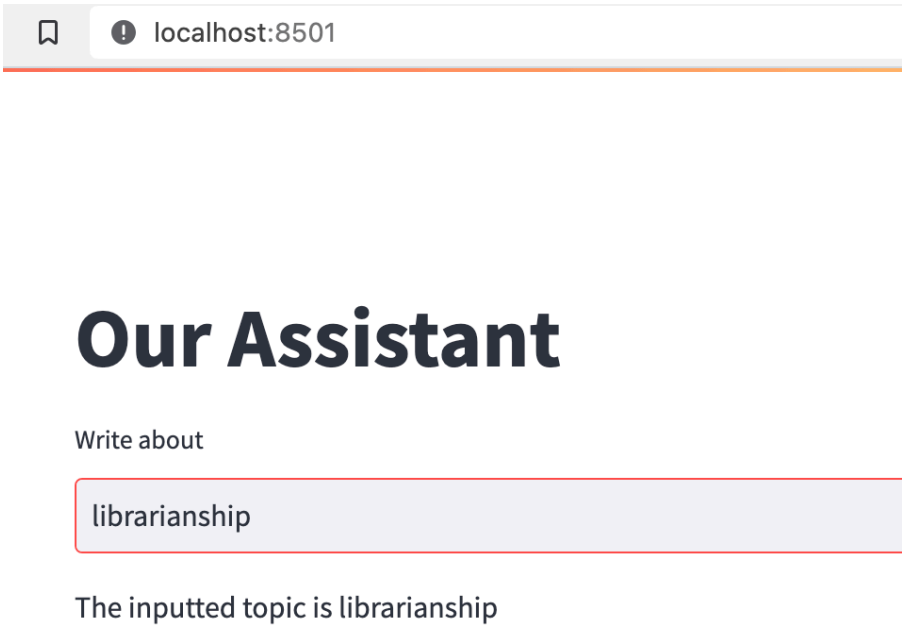


Figure 31. Assistant Showing Input of 'librarianship'.

Adding an LLM

We can add an LLM (and check its response):

```
# % pip install streamlit langchain openAI wikipedia

import streamlit as st
import langchain
import os
from langchain.chat_models import ChatOpenAI

os.environ["OPENAI_API_KEY"] = '<INSERT YOUR API KEY HERE>'
# insert your API_TOKEN here

#Streamlit display
st.title ('Our Assistant')
input = st.text_input ('Write about')

#Get ourselves an llm
llm = ChatOpenAI(model_name="gpt-3.5-turbo")

#Testing we have input and letting the llm respond
if input :
    response = llm.call_as_llm(prompt)
    st.write(response)
```

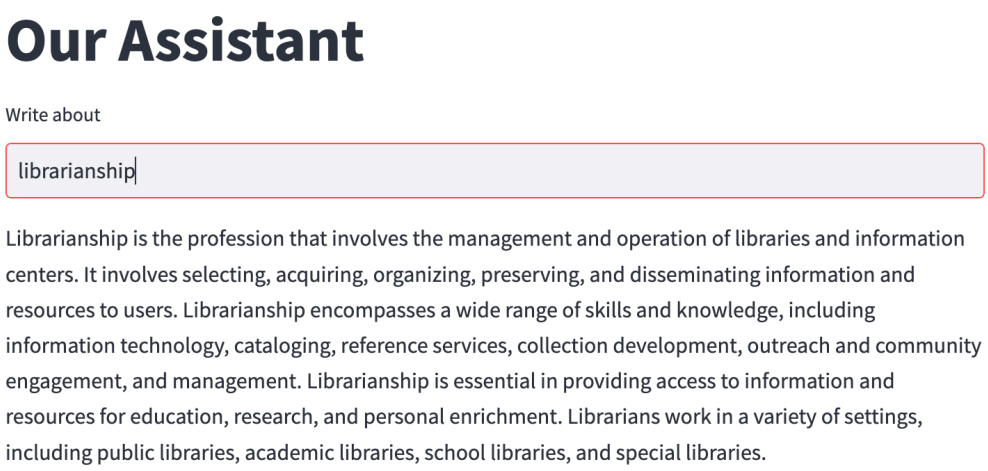


Figure 32. Assistant Showing Response to Input of 'librarianship'.

Prompt Templates

Prompts are hard to get right. So, once you have a recipe that works it is a good idea to make a template out of it and use that. Templates have zero or more variables and some boilerplate. Then values for the variables will be inserted into the template to produce the actual prompt. We will make a template out of:

> "Write one paragraph in the style of Emily Dickinson on the topic of {topic}"

```
# % pip install streamlit langchain openAI wikipedia

import streamlit as st
import langchain
import os
from langchain.chat_models import ChatOpenAI
from langchain import PromptTemplate

os.environ["OPENAI_API_KEY"] = <INSERT YOUR API KEY HERE>
#insert your API_TOKEN here. Sorry you can't use mine

#Streamlit display
st.title ('Our Assistant')
input = st.text_input ('Write about', key='about')


#Get ourselves an llm
llm = ChatOpenAI(model_name="gpt-3.5-turbo")

#Create a template
template = "Write one paragraph in the style of Emily
Dickinson on the topic of {topic}"
```

```
prompt_template = PromptTemplate.from_template(template)

#Testing we have input and letting the llm respond
if input :
    real_prompt = prompt_template.format(topic=input) #put
the  input into the template
    st.write("Real prompt:", real_prompt)
#check
    response = llm.call_as_llm(real_prompt)      #ask llm
    st.write(response)
```

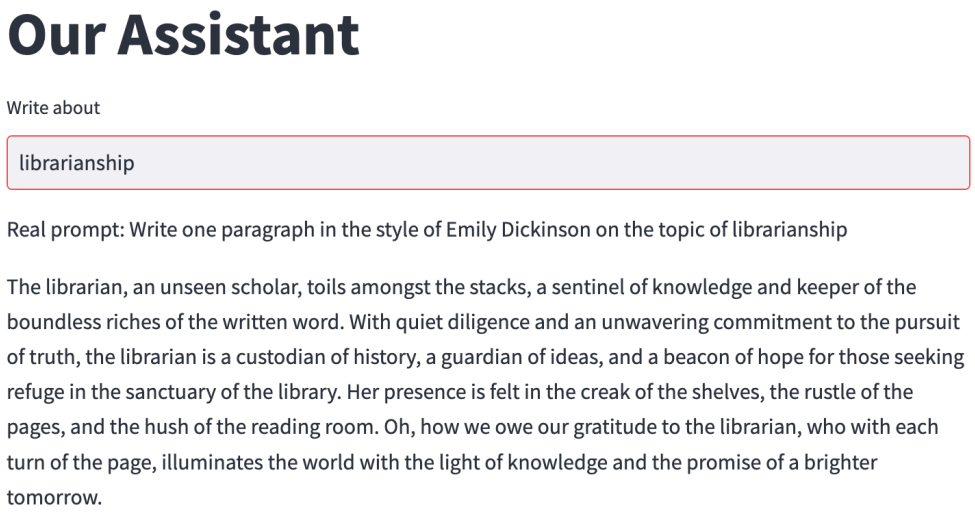


Figure 33. Assistant Showing Response to Input of 'librarianship' in the style of Emily Dickinson.

Of course, the LLM in general will likely give a different response every time it is asked. Here is an alternative that it offered:

> The librarian, with her quiet grace, is the keeper of the written word. She moves among the stacks, a gentle guardian of knowledge. With each book she takes down, a new world unfolds before her eyes. She is a seeker of truth, a lover of language, and a protector of ideas. Her hands are calloused from years of handling pages, but her spirit remains unbroken. She is the keeper of

secrets, the purveyor of dreams, and the champion of learning. The librarian is a treasure, a rare gem of wisdom and wonder.

Document Embeddings

LLMs will be trained possibly on a large portion of the public facing Internet. But there are occasions when you might want to supplement this. As examples, an academic might want regularly to access and query all her own research publications and lectures, a library might have a special collection of materials in need of LLM processing.

If the new content is brief— say a single newspaper article— it could be prepended to the prompt. But the prompts can only be short, say a few thousand 'tokens', so this idea is clunky and is not going to work in general. What is needed is for all the new documents to be processed into embeddings (i.e. lists of numbers) and those lists to be stored in a vector database. The LLM will then augment its background by using the database.

Python and LangChain have a number of tools to get the documents in the right form in the first place (e.g. Python Beautiful Soup can scrape web pages, Python PyPDF library can extract texts from pdfs).

Just as a brief example here. Chapter 1 of this text has been printed as a pdf and put in a directory called SamplePdfs. That is our library (we needed a sample pdf not covered by copyright). We will convert it to embeddings (numbers), and store those in a vector database. We will use FAISS

(Facebook AI Similarity Search) database. Then we can ask questions of it. Here is an example

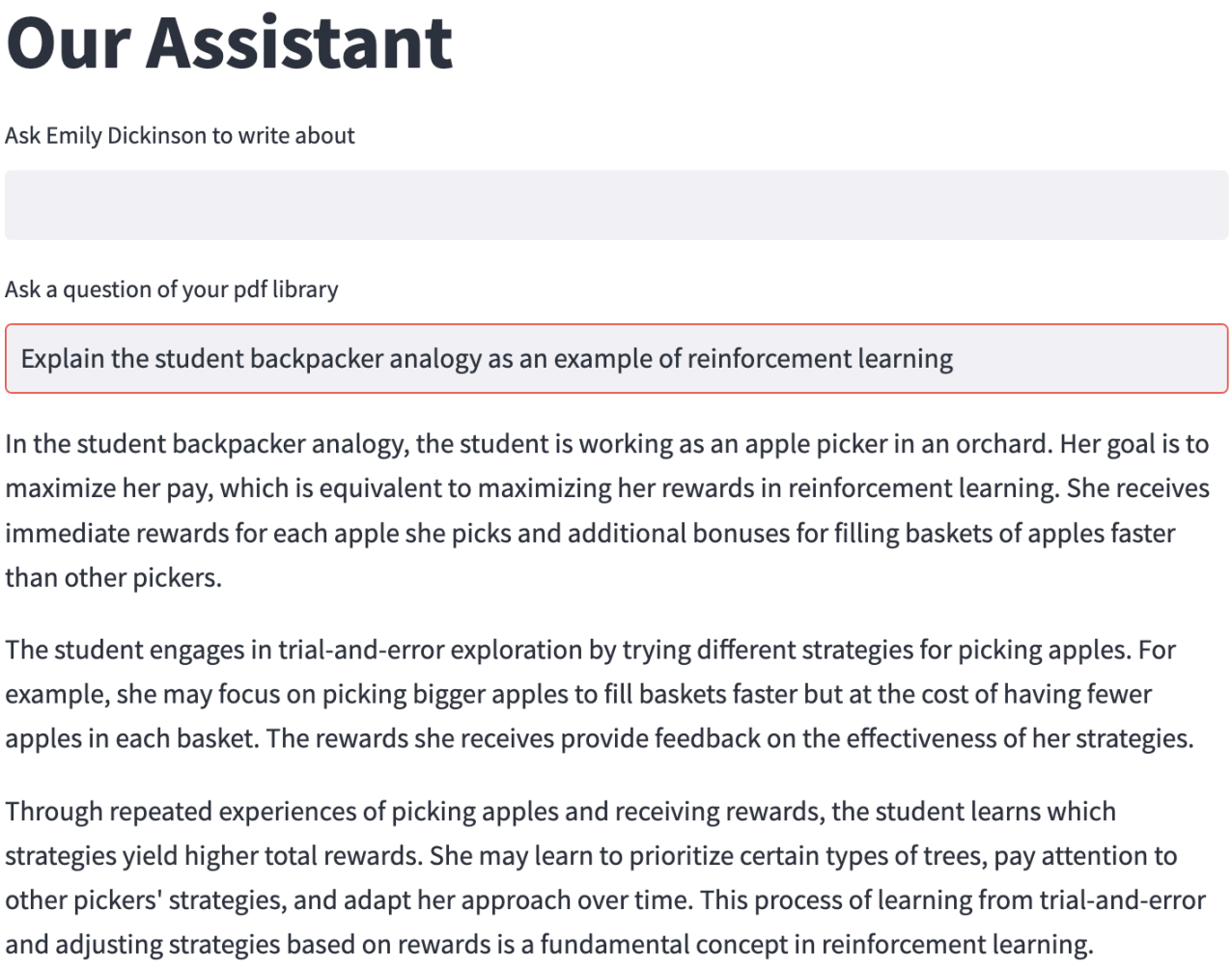


Figure 34. Assistant Explaining the Backpacker Analogy.

GPT-3.5-Turbo is answering our question, not from what it knows from its training but from our library. In this simple case, the library has a single 47 page pdf in it. But it easily could have every piece of research that a scholar has ever written.

Separately, we could not resist hearing from Emily Dickinson again:

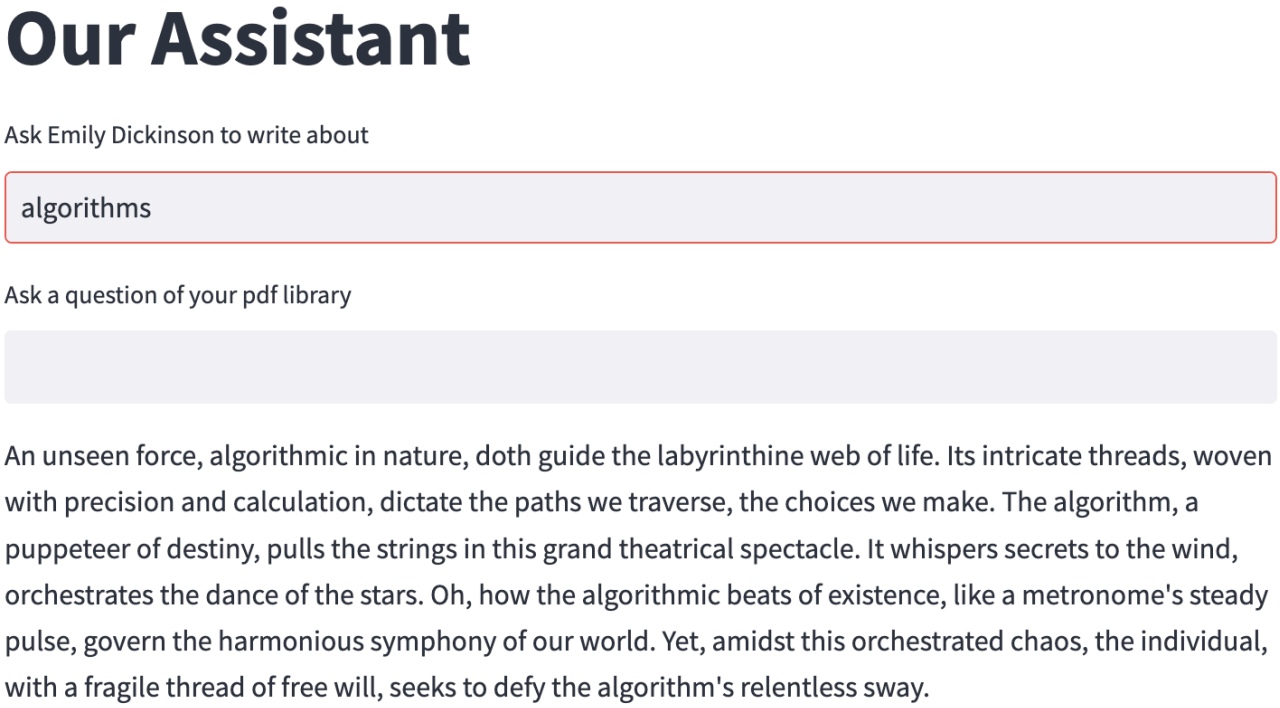


Figure 35. 'Emily Dickinson' Explaining Algorithms.

The python code for our assistant is as follows. [The code here is written in a very idiosyncratic style, please do not take it as anything to be admired or copied.]

```
# % pip install streamlit langchain openAI pypdf

import streamlit as st
import langchain
import os
from langchain.chat_models import ChatOpenAI
from langchain import PromptTemplate

os.environ["OPENAI_API_KEY"] = <INSERT YOUR API KEY HERE>

#Streamlit display. Two entry possibilities
st.title ('Our Assistant')
```

```
input = st.text_input ('Ask Emily Dickinson to write about',
key='about')
question = st.text_input ('Ask a question of your pdf library',
key='question')

#Get ourselves an llm
llm = ChatOpenAI(model_name="gpt-3.5-turbo")

#Create a template
template = "Write one paragraph in the style of Emily Dickinson
on the topic of {topic}"

prompt_template = PromptTemplate.from_template(template)

#Testing we have input and letting the llm respond to the
templated version
if input :
    real_prompt = prompt_template.format(topic=input) #put the
input into the template
    response = llm.call_as_llm(real_prompt)           #ask llm
    st.write(response)                                #write the
response


########### Going to load the pdfs in a directory (use either
unstructured pdf or pypdf)

# pip install langchain unstructured openai tiktoken pypdf

#loading the pdf docs from the SamplePdfs directory

from langchain.document_loaders import PyPDFDirectoryLoader
pdf_folder_path='SamplePdfs/'
loader = PyPDFDirectoryLoader(pdf_folder_path)
docs = loader.load()

#try FAISS (Facebook AI Similarity Search) as our database

#pip install faiss-cpu
from langchain.vectorstores import FAISS

########### Making the embeddings

from langchain.embeddings import OpenAIEmbeddings

embeddings = OpenAIEmbeddings()
```

```
#create the vector store from the embeddings of our library, to
use as the index
db = FAISS.from_documents(docs, embeddings)

#get a 'retriever' that will ask quesions of our database
from langchain.chains import RetrievalQA
retriever = db.as_retriever()
qa = RetrievalQA.from_chain_type( llm=llm,
                                  chain_type="stuff",
                                  retriever=retriever,
                                  return_source_documents=True)


#if there is a question, answer it from our library
if question:
    answer = qa({"query": question})
    st.write(answer['result'])
```

[Ben Dickson has a discussion of using your own documents in (Dickson 2023)]

There is Lots More

Needless to say. Let us hope there is enough here to stimulate your interest.

A Useful Resource

Nicholas Renotte's excellent *LangChain Crash Course: Build a AutoGPT app in 25 minutes!* is a useful resource. His accompanying video is hosted at https://www.youtube.com/watch?v=MlK6SIjcjE8 and the code he uses is available from https://github.com/nicknochnack/Langchain-Crash-Course/blob/main/app.py.

B.4.3 LangChain Using Jupyter

If instead, or as well, you may wish to use Jupyter….

A good way to get Jupyter notebooks is to use an Anaconda installation. Go to https://anaconda.com and download a free installation (probably from https://www.anaconda.com/download). Create yourself a directory (folder) for your Jupyter notebooks. Launch Anaconda-Navigator, that will give you

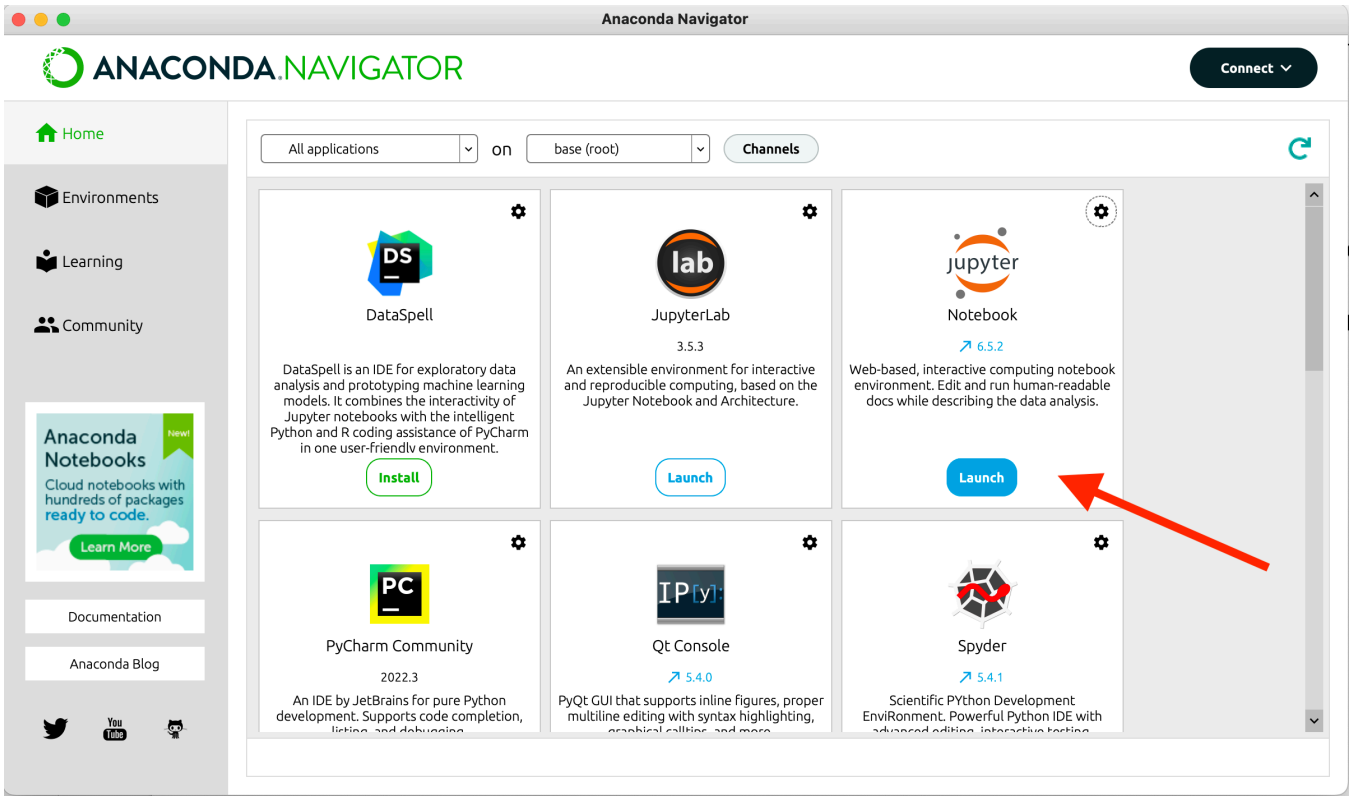


Figure 36. Anaconda Navigator.

Launch Jupyter Notebook. That will open a web page in a browser, looking similar to this

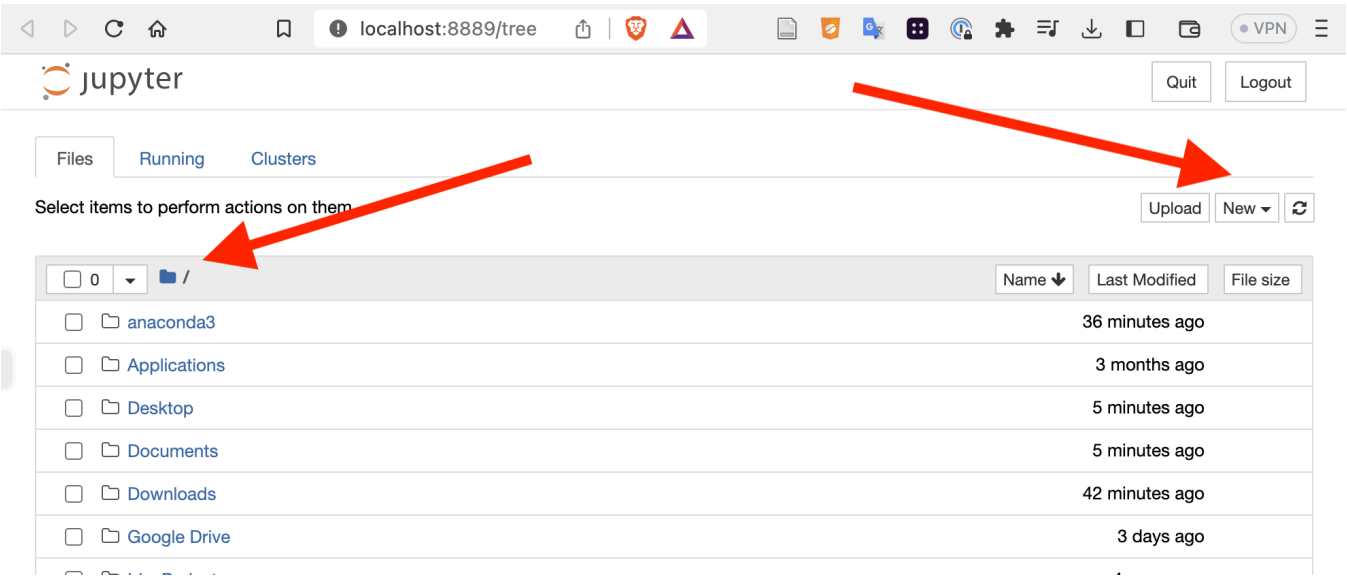


Figure 37. Jupyter Display of Directory Structure.

Navigate to the folder you are going to use for your Notebooks and, off the New button, create a new notebook. Type into the first cell `print ("Hello World")`.

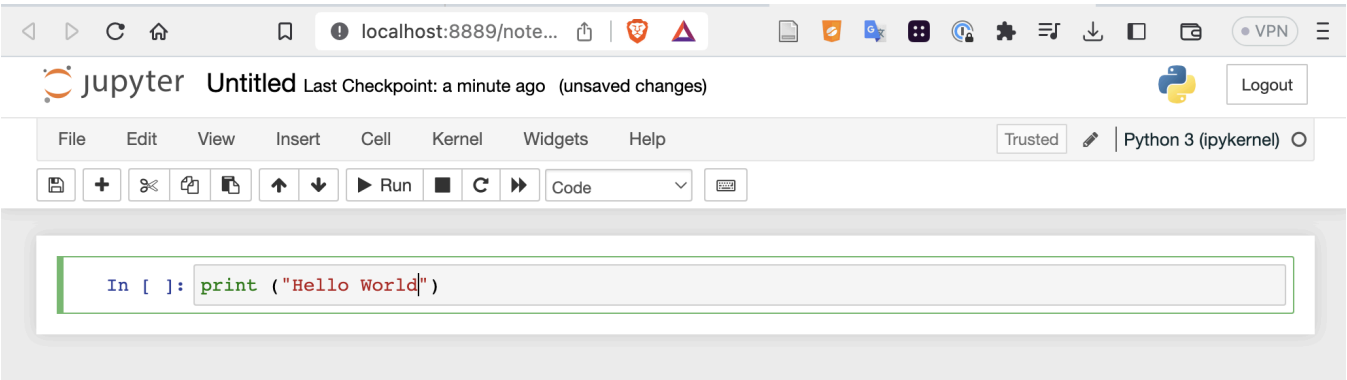


Figure 38. A Jupyter Notebook Being Given the Input 'print ("Hello World")'

Then either click Run or type shift-enter. This will 'run' or evaluate the cell

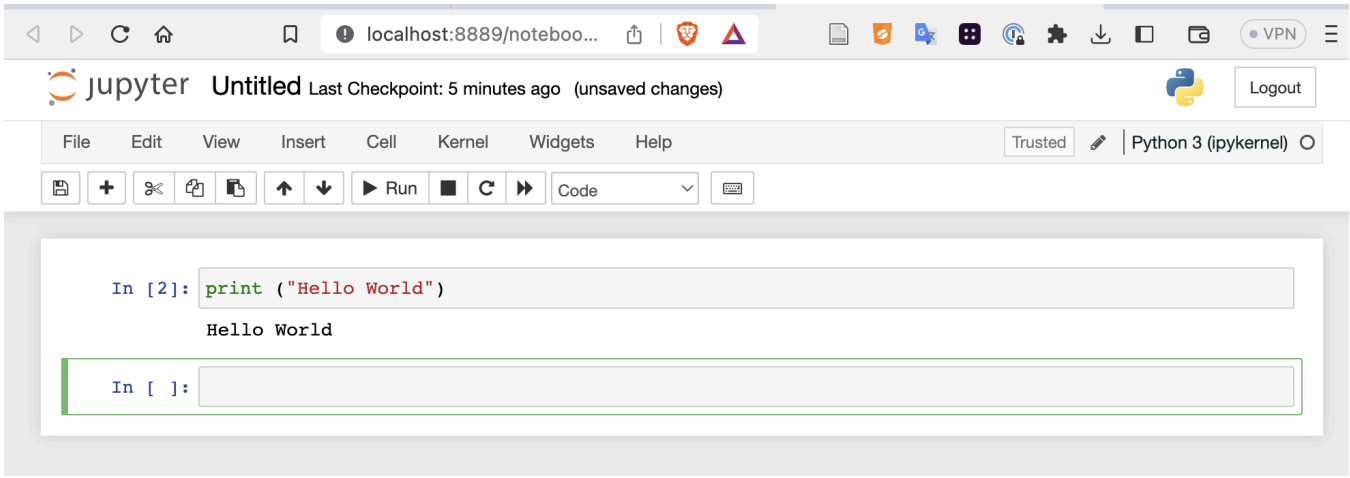


Figure 39. Jupyter Evaluating the Cell Displayed in Figure 38.

Now you have Jupyter notebooks running Python. We do not need expertise in either Jupyter or Python, but at least some level of comfort is required. If you feel you do not have that, there are a myriad of excellent resources on You Tube and the Web (for example, Jupyter's own documentation or Codecademy's How to Use Jupyter Notebooks.)

The package installer pip can be used within Jupyter notebooks, but care is needed. The problem is that Jupyter is so powerful that it might be running all sorts of instances of Python in all sorts of places. You need to be sure that the package is installed with the right Python. Use the following code in a cell of your notebook and evaluate it:

```
import sys
!{sys.executable} -m pip install <insert the package you
require here>
```

The `{sys.executable}` fragment picks up the Python that is running.

[A Jupyter notebook has the suffix .ipnyb.] We will have an interest in Greg Kamradt's Cookbook (Langchain Cookbook Part 1 - Fundamentals Part 1 .ipnyb and Langchain Cookbook Part 2 - Fundamentals Part 2 .ipnyb). A way to get this is to go to https://github.com/gkamradt/langchain-tutorials click on the green code button and download the ZIP compression of all the files. (Not everyone is comfortable downloading ZIP files from the Internet. If you have concerns, and caution is definitely in order, just do not do it.) Assuming you have the ZIP, expand it into the folder you are using for your Jupyter LangChain files.

B.4.4 Resources for LangChain using Jupyter

For LangChain in the context of Jupyter, we are going to suggest two sources: Greg Kamradt's excellent *The LangChain Cookbook - Beginner Guide To 7 Essential Concepts* (Kamradt [2023] 2023), and James Briggs and Francisco Ingham's also excellent *LangChain AI Handbook* (Briggs and Ingham 2022). The latter comes from the company Pinecone, which hosts vector databases in the cloud (an important part of infrastructure for NLP).

For the Kamradt, there is a video and github resources (which we have downloaded as a zip):

- The LangChain Cookbook - Beginner Guide To 7 Essential Concepts [video] and there is a second and other videos at (Kamradt 2023)
- The LangChain Cookbook [Jupyter] This uses OpenAI.

For the Briggs and Ingham, there are 12 videos from James Briggs, Jupyter notebooks and a book.

- Getting Started with GPT-3 vs. Open Source LLMs - LangChain #1 [video] This uses OpenAI and Hugging Face, and it links to the other videos.
- The Jupyter code notebooks for this are at https://github.com/pinecone-io/examples/tree/master/generation/langchain/handbook and the notebook for the first video is 00-langchain-intro.ipynb (you can download this by clicking on the download symbol, then you can open it in your running Jupyter).
- The handbook is (Briggs and Ingham 2022).

B.5 Annotated Resources for Appendix B

Costa, Ricky. “ChatGPT Cheat Sheet.” Https://Neuralmagic.Com/, 2023. https://www.kdnuggets.com/publications/sheets/ChatGPT_Cheatsheet_Costa.pdf. (Costa 2023)

Deepankar, and Florian. “PromptPerfect - Elevate Your Prompts to Perfection with AI Prompt Engineering,” 2023. https://promptperfect.jina.ai/. (Deepankar and Florian 2023)

Huyen, Chip. “Building LLM Applications for Production,” 2023. https://huyenchip.com/2023/04/11/llm-engineering.html. (Huyen 2023) This is good. Huyen writes ' It’s easy to make something cool with LLMs, but very hard to make something production-ready with them.'

Mollick, Ethan. “Working with AI: Two Paths to Prompting,” 2023. https://www.oneusefulthing.org/p/working-with-ai-two-paths-to-prompting. (Mollick 2023)

Monigatti, Leonie. Getting Started with LangChain: A Beginner’s Guide to Building LLM-Powered Applications (Monigatti 2023)

OpenAI. “Guide to Prompt Engineering,” 2023. https://platform.openai.com/docs/guides/prompt-engineering. (OpenAI 2023c)

Appendix C: Two Important Methodological Points

There is a conventional symbolization that will help here with probabilities. We will write p(A) to mean the probability of A where A is some sentence, for example, p(it is raining). Then there is the notion of *conditional probability* written p(A|B), where A and B are both sentences, and this is read 'the probability of A given B' or 'the probability of A given the condition B', for example p(Jane gets wet|it is raining) which would be read 'the probability of Jane getting wet given than it is raining'.

C.1 False Positives and False Negatives

> In medical testing, and more generally in binary classification, a false positive is an error in data reporting in which a test result improperly indicates presence of a condition, such as a disease (the result is positive), when in reality it is not, while a false negative is an error in which a test result improperly indicates no presence of a condition (the result is negative), when in reality it is present. (Wikipedia 2022d)

Aesop's fable of the Shepherd Boy who cried 'Wolf' provides a classical partial illustration of this. We can modify the fable to become a full example. In Aesop, the boy is supposed to cry 'Wolf' when there is a wolf. So, the condition is the presence of a wolf, and the positive test for this is the boy's cry of 'Wolf'. Now, as we all know, in the fable, the boy become bored and lonely, and started crying 'Wolf' even though there was no wolf. These cries are all **false positives** (the test is positive but the condition does not exist). The villagers responded to these false positives, several

times over, rushing to be with the boy and to protect their flock. Later, a wolf actually did appear, and the boy cried 'Wolf'. This cry is a **true positive** (the test is positive and the condition does exist). However, the villagers did not respond i.e. they ignored a true positive (largely because they were tired of false positives and mistook a true one for a false one). Here is our modification. All of us, the villagers and ourselves, know that not being told that there is a wolf present is not the same as being told that there is not a wolf present. So, the villagers, desirous of peace of mind, set up the situation differently going forward. Apparently, the wolves in that region were crepuscular. That means that they hunt at dawn and a dusk. They gave the Shepherd Boy a second task: he also had the job of crying 'No wolf' once at dawn and at dusk, on the condition that there was indeed no wolf. Initially, the Shepherd Boy was conscientious with the second task. He made the appropriate cries in the absence of wolves. These cries are all **true negatives**. They are supposed to indicate the absence of a condition and they do exactly that. But, we know, the boy was a bit of a larrikin. Sure enough, one dusk, a wolf appeared, and yet the boy cried 'No wolf'. This cry would be a **false negative** (the test is negative, but yet the condition does exist).

During a 24 hour period, the villagers might a) hear nothing (in which case, there would have been a failure in duties), b) hear cries of 'Wolf' or 'No Wolf' at different times, dawn or dusk (these individually could be true or false positives, or true or false negatives), or c) hear cries of 'Wolf' or 'No Wolf' at the same time (in which case, they would know that either the 'Wolf' cry was a false positive or the 'No Wolf' cry was a false negative).

The various relations here between condition and test result can be expressed as conditional probabilities. We will relax our terminological conventions a little and just write ‘WolfCry’ and ‘NoWolfCry’ for the cries and ‘WolfPresent’ and ‘NoWolfPresent’ for the conditions. Then the probabilities are

p(WolfCry|WolfPresent)	True positive
p(WolfCry|NoWolfPresent)	False positive
p(NoWolfCry|NoWolfPresent)	True negative
p(NoWolfCry|WolfPresent)	False negative

The general theory here finds its greatest application perhaps in medicine. We all know of false positives, negatives, and the like, in the context of Covid tests. Almost all real world tests do have false positives, false negatives, etc. Typical values for these probabilities might be around 0.05 (roughly, 1 in 20 results is a false positive or false negative). Were the probabilities to be much higher than this, the tests would be regarded as being unsatisfactory.

As we will see shortly, the fact that a test seems to 'indicate' that a person does or does not have a disease with a certain probability does not actually means that the probability in question is indeed the probability of the person actually having the disease. There is another factor.

C.2 The Base-Rate Fallacy

The Base-Rate Fallacy, sometimes known under the heading ‘Harvard Medical School Test’, is probably the most common case of probabilistic

reasoning where in real life almost everyone is tempted to reason incorrectly.

Say there is some dread disease— Lurgi— and there is a test for the disease. This test is very good. So good, in fact, that if anyone actually has the Lurgi then there is a 0.95 probability that the test will show positive (and a 0.05 probability, when the person actually has the Lurgi, that the test will say that they do not have it— the 'false negatives') i.e.

p (PositiveTest | Lurgi) = 0.95 or 95%
p (NegativeTest | Lurgi) = 0.05 or 5%

Now assume we test John Smith, and sad to say, the test is positive. So, does John Smith have the Lurgi? Is it probable that he has the Lurgi? Do we know anything at all about whether John Smith has the Lurgi (on the basis of this test and its result alone)?

We know two pieces of information

p (PositiveTest | Lurgi) = 0.95 or 95%
John Smith has tested positive.

and we are trying to find out John Smith's status viz-a-viz Lurgi.
What we would like to know, or need to know, first off, is this

p (Lurgi | PositiveTest) = ?

but there is no way of reasoning from

p (PositiveTest | Lurgi)

alone, to

p (Lurgi | PositiveTest)

One aspect of the difficulty is that we do not know the rate for the false positives. (Another is that we do not know the background rate, or base rate, for Lurgi. We will get to that shortly). Conclusion: we know nothing about John Smith's status viz-a-viz Lurgi.

Now, let us allow us to have information on the false positives. Say

p(PositiveTest|~Lurgi)=0.1

We now know a) the probabilities for the true positives (0.95) and for the false positives (0.1) and b) that John Smith has tested positive. Do we know whether it is probable that John Smith has Lurgi. What we are being tempted with here is what is known as the *base-rate fallacy* (Amos Tversky and Kahneman 1982).

Many of the misleading (or trick) examples of this are set up in the same way. There is a very low background probability of something, say of a person having Lurgi, which we will set for this example as being 0.01. And there is some sort of test for the condition of having Lurgi, which is pretty good, say p(PositiveTest|Lurgi) = 0.95. But the test also gives some false positives (that is to say, it occasionally indicates that a person has Lurgi when they do not have it), say p(PositiveTest|~Lurgi)=0.1. Then we are

told the following story and asked the following question. A person goes in and is tested for Lurgi and the test is positive, is the probable that the person has Lurgi? Most of us say that it is, whereas, in fact, it is very unlikely. A good way to see this (and how Bayes' Theorem applies) is to re-tell the story in terms of "natural" frequencies. In this story, you live in a town of 10000 people and 100 of them have Lurgi. Everybody is tested for Lurgi. Of those hundred people with Lurgi, 95 test positive. Of those 9,900 without Lurgi, 990 test positive for Lurgi (the false positives). You test positive for Lurgi. Is it likely that you have Lurgi? Well, you have a 95/990 i.e. about a 1 in 11 chance of having it, and about a 10 in 11 chance of not having it. You probably do not have it. The correct reasoning here is an instance of Bayes' Theorem, in the form

$$p(\text{Lurgi}|\text{PositiveTest}) = \frac{p(\text{positive} \mid \text{Lurgi}) \times p(\text{Lurgi})}{p(\text{positive}|\text{Lurgi})p(\text{Lurgi})+p(\text{positive}|\sim\text{Lurgi})p(\sim\text{Lurgi})}$$

With numbers

$$p(\text{Lurgi}|\text{PositiveTest}) = \frac{.95 \times .01}{.95x.01+.1x.99}$$

=>

$$p(\text{Lurgi}|\text{PositiveTest}) = \frac{.0095}{.0095 + .099}$$

What we are tempted to do when reasoning badly is a) to focus on how good the test is when giving positive results from positive cases, and b) ignoring the background rate (how rare the disease is, *simpliciter*). And

what we need to do is a) take the false positives into account, check how often the test gives a positive result from a negative case, and b) remember the background rate (then, roughly: if the disease is rare, and the test can give false positives, the probability is that a positive is a false positive).

Consider this. We have a test that is pretty good in that it usually comes out positive for those that have the disease, but it does produce some false positives, perhaps 5%. This means that if you test 100 people, who do not have the disease, 5 of them might test positive. It also means that if you test 100 million people, who do not have the disease, 5 million of them might test positive. That is quite a lot! So, if you screened the entire population of the US (say 300 million) you might have 15 million false positives. Now if the disease is very rare in the population (for example, folk in the US having Ebola, which might be 1,2, or 3 people only). If you test someone for in the US for Ebola (with one of those 5% false positive tests above), and they test positive, it is much more likely that they are a false positive (and they don't have Ebola) than it is that they have Ebola. Thinking otherwise is the so-called base rate fallacy.

[Experts will know that, strictly speaking, parts of the 'natural frequencies' explanation are not entirely correct in full detail. We will not worry about that here.]

C.3 Annotated Readings for Appendix C

Howson, Colin, and Peter Urbach. Scientific Reasoning : The Bayesian Approach. 3rd ed. Chicago: Open Court, 2006. (Howson and Urbach 2006) Bayesian techniques are widespread in modern science (and in machine learning). The Howson and

Urbach book is excellent. While it is well written and an engaging read, it may be a little advanced for us.

Appendix D: Causal Diagrams

D.1 Causation and Correlation

It may be wise to say a word or two about causality. Philosophers have studied causality for thousands of years. They have made progress, but their theories are way too complex for us. The computer scientist Judea Pearl introduced a way of thinking about causality that was suitable for reasoning about causality in the setting of artificial intelligence and machine learning (Pearl 2009b). What was needed here is some principled way of understanding correlation and causation and their differences. Here is a proposal following from the wider work of Pearl and his intellectual colleagues. In statements like:

> Taking aspirins causes relief from headaches.

there are three features of interest. There is a direction, a direction in time. The earlier taking of aspirins produces, brings about, or 'causes' the later relief from headaches. The later relief from headaches does not produce, bring about or 'cause' the earlier taking of aspirins. Second, there is an association or correlation between the cause and the effect. In the example case, there is a regularity between taking aspirins and relief from headaches. This regularity is not an absolute guarantee. For one reason or another, the taking of aspirins does not always relieve headaches on all occasions. However, the taking of aspirins does increase the chance, or likelihood, or probability, that the headaches will be relieved. Thirdly, there is what might be called an intervention or counterfactual factor. Often with

causality, we have the ability to intervene or produce or change or manipulate the cause in an attempt to manipulate the effect. This is a great and desirable feature. Were our teenage child to have a headache, we could give them an aspirin and this may well provide relief. Similarly, here we can reason counterfactually. If the child in fact had not been given an aspirin, we might make the consoling observation 'you know, an aspirin would have helped you'. Plain correlation, without causation, does *not* have a direction, *does*, or *can*, involve probabilities, does *not* give us the ability to manipulate outcomes, and does *not* support counterfactuals. Cirrhosis of the liver is caused by drinking alcohol. Smoking is correlated with cirrhosis of the liver (among the population to date). Current smokers and non-smokers have different probabilities of having cirrhosis of the liver. Those with cirrhosis of the liver and those without cirrhosis of the liver have different probabilities of being smokers. But an intervention that stops you smoking, if you are a smoker, does not change your probability of getting cirrhosis of the liver (provided all other factors are unchanged, in particular whether you drink or not). Smoking is correlated with cirrhosis of the liver, but it does not cause cirrhosis of the liver.

In sum, with causality, there is a direction, changing of probability, and the possibility of interventions and counterfactuals.

There can be causal talk, correlation talk, and 'weaselly' talk (which is talk intending to make the reader think of causality.) Nick Huntington-Klein gives useful examples of the words and phrases in use here:

> What are some of these words?

> We can say that X causes Y by saying: X causes Y, X affects Y, the effect of X on Y, X increases/decreases Y, X changes Y, X leads to Y, X determines Y, X triggers Y, X improves Y, X is responsible for Y, and so on...
>
> We can say that X and Y are related without implying causality by saying X and Y: are associated, are correlated, are related, tend to occur together, tend not to occur together, go together, and so on...
>
> If some weaselly writer ... doesn't want to *say* causality but does want the reader to *hear* it, they might say: X is linked to Y, X is followed by Y, X has ramifications for Y, X predicts Y , people who X are more likely to Y , Y happens as X happens, and many others.
>
> Knowing these terms can help you interpret what scientific studies are really saying, and when someone might be trying to pull one over on you (Huntington-Klein 2022).

D.2 Causal Diagrams

Researchers in ML, and, indeed, in causality and statistics in general, often employ causal diagrams (Pearl 2009a; Scheines 1997; Pearl 1995). These are useful in many settings, in particular with being assured that predictions have genuine substance, and with addressing questions of bias and fairness. When we can identify causes in a system, two possibilities open up. If we can also manipulate or adjust or change the causes, we may be able to change the effects (and this may be very desirable). Separately, we can start reasoning counterfactually. That is, thinking what would, or might happen, were the causes to be changed (and this counterfactual analysis may give us insight on bias and fairness, among other things).

Causal diagrams use variables and arrows. For example, if we think, as a causal model, that smoking causes lung cancer, the following diagram might be suitable:



Figure 40. A Causal Diagram Showing Smoking Causing Lung Cancer.

The arrow indicates our views on causality (in this case, that smoking causes lung cancer— that smoking is a *direct* cause of lung cancer). The arrow is an arrow of causality. The causality flows 'downstream' from the smoking to the lung cancer. Sometimes the direct cause, or tail of the arrow, is called the 'parent', then the adjacent variable that the head of the arrow attaches to would be the 'child'.

Such diagrams have two roles. As far as causality is concerned, there is no data involved. We have produced this model out of thin air, or out of our ideas, theories, conjectures, or background knowledge. The causality here is not deterministic. It does not mean that every single smoker gets lung cancer as a piece of inexorable mechanistic clockwork. Rather, it is about the category or type or class of smokers— that at least some of them are caused to get lung cancer by their smoking (other things being equal). If this diagram is correct, the second role will come into play, and that is that there will be a statistical connection, or correlation, or dependence, or association, between smoking and lung cancer. Data is involved here. We

would expect to see a correlation between the folk who smoke and the folk who get lung cancer. The diagram says nothing at all about what it is to be a smoker— whether the values here are just 'Yes' or 'No' or whether there are grades or degrees of being a smoker. Similarly, it says nothing at all about the values for the variable lung cancer. Also, the diagram says nothing about what the causal effect is. We know full well that being a smoker *increases* the probability of getting lung cancer. But as far as the semantics of the diagram is concerned, the arrow just asserts that there is some causal effect, or causal association between the two variables. Smoking may increase lung cancer, it may decrease lung cancer, the arrow in the diagram is agnostic on this.

Arrows show causal connection. Lack of arrows show (presumed) absence of causal connection. We have only two variables in our diagram. More complex diagrams may have many more variables. We have to be explicit with our commitments— put arrows between variables if our model assumes a causal connection, omit arrows where we assume no connection. Obviously, for example, in our simple case, something may cause smoking, and lung cancer itself may cause other effects. But a causal diagram does not need to contain the world history, or the world future. There just needs to be enough variables for the problem at hand.

There is a requirement or constraint on the variables and arrows. If any two variables have a common cause— that is, another variable that causes both of them— that variable, that common cause, and its arrows, need to be in the diagram. [This requirement is known as the *Causal Markov Condition*.] We have only two variables in our diagram, so only two to check (i.e.

smoking and lung cancer). But suppose we theorize that there is a gene that causes smoking and, also, that gene causes lung cancer (the gene does, by itself). Then, to be a causal diagram— to satisfy the Causal Markov Condition— our diagram would need to be modified to:

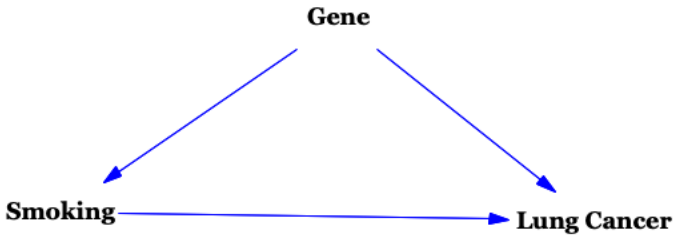


Figure 41. Smoking Causing Lung Cancer And a Gene Causing Both of These Conditions.

[In fact, we are not here going to assume that there is such a gene, so the original diagram, with just two variables is a causal diagram as it stands. It satisfies the Causal Markov Condition.]

There are three components, or building blocks, or modules, that might occur in a causal diagram: *chains*, *forks*, and *collisions*.

To introduce chains, we need *paths* and paths are sequences of adjacent arrows. If the arrows in a path connect head-to-tail, the path is a *directed* path, otherwise the path is an *undirected* path. A *chain*, or *causal chain*, is a directed path between variables. In the diagram:



Figure 42. A Causal Chain From Smoking to Lung Cancer.

There is just the one chain: **Smoking->Lung Cancer**. More detail can be added between the variables for smoking and the lung cancer. For example, there might be cell mutation and it may be valuable to include an (intermediary) variable for that. A more complete diagram might be:

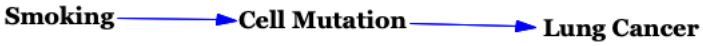


Figure 43. A Causal Chain From Smoking to Lung Cancer Mediated by Cell Mutation.

In this diagram there are three (causal) chains: **Smoking->Cell Mutation, Cell Mutation->Lung Cancer, Smoking->Cell Mutation->Lung Cancer**. Interest will likely be with causal chain between smoking and cancer, and if indeed smoking causes cancer there will be correlation between the two. There also will be a correlation between smoking and cell mutation and cell mutation and lung cancer.

A *fork* is where there is a common cause of two variables. If we think, as a causal model, that smoking causes both lung cancer and yellow stains on a smoker's fingers, the following diagram might be suitable:

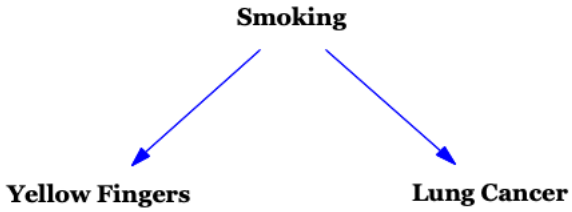


Figure 44. A Fork from Smoking to Yellow Fingers and Lung Cancer.

In this, smoking is a common cause of both yellow fingers and lung cancer. This, a common cause, is a *fork*. Forks need care where correlations are concerned. Smoking will be correlated with yellow fingers, smoking will be correlated with lung cancer, and *yellow fingers will be correlated with lung cancer*. There is a little more that can be said. There is the notion of *conditioning* and a simple explanation of conditioning is that it is knowing or fixing the value of a variable. Suppose the smoking variable can have two values only: being a smoker, or not being a smoker. Consider just non-smokers. Some of them will have yellow fingers. Few or none of them will have lung cancer. But now there will be *no* correlation between the yellow fingers and lung cancer. It is the smoking that causes lung cancer, but *none* of the people in the group are smokers. Equally, consider just smokers. Some of them will have yellow fingers. Some of them will have lung cancer. Again, there will be *no* correlation between the yellow fingers and lung cancer. It is the smoking that causes lung cancer, but *all* of the people in the group are smokers. So, if we conditionalize on the smoking variable, there is no correlation between yellow fingers and lung cancer. In sum here, where there is a common cause, a fork, there is correlation (or dependence)

between the effects. But if the analysis conditions on the common cause there is no conditional correlation (or conditional dependence) between the effects.

A *collision* is when there is a common effect of two different causes. Smoking is not the only action that causes lung cancer. Exposure to asbestos can cause lung cancer. We might not be especially interested in the asbestos cause. But if we are, we might produce a diagram similar to this:

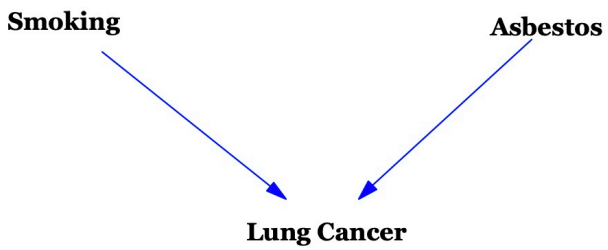


Figure 45. A Fork from Smoking and Asbestos to Lung Cancer.

Consider the path **Smoking -> Lung Cancer <- Asbestos.** This is an undirected path, and the path enters and leaves the same variable (here Lung Cancer) by arrow heads. This means that the variable (here Lung Cancer) is a *collider* in this path. Paths that have colliders in them are *closed* or *inactive*. Paths that do not are *open* or *active*. Directed paths tell of causality, of causal chains. So, at the level of causality, we know, or assume, that smoking causes lung cancer and asbestos causes lung cancer. But we also know, or assume, that smoking does not cause exposure to asbestos nor does exposure to asbestos cause smoking. At the level of statistics, we would expect there to be a correlation between smoking and

lung cancer and a correlation between asbestos and lung cancer, but no correlation between smoking and asbestos. Actually, where colliders are concerned again there is a little more to be said. To explain this, it is useful to have different examples of colliders. The first is for a kitchen light. There is a door from outside to the kitchen, and a door from the kitchen to the rest of the interior of the house. Each of these doors has a kitchen light switch near it. These switches turn the single kitchen light on or off:

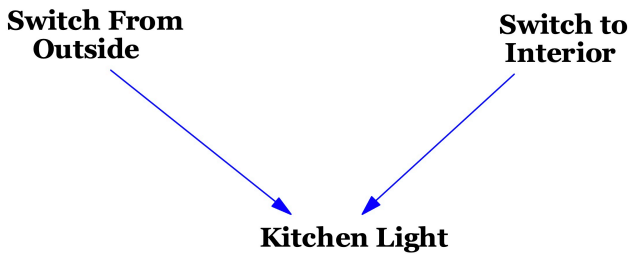


Figure 46. A Collider Between Two Switches and a Light.

In the path **Switch From Outside -> Kitchen Light <- Switch To Interior** there is a collider. There is causality, and correlations, between the switches and the light, but not between the switches. Whether one switch is on (or off) is entirely independent of whether the other switch is on (or off). Looking at one of the switches alone will tell you nothing about the other switch. But let us conditionalize on the **Kitchen Light** i.e. permit information as to whether the light is on or off. Now there will be correlation between the switches (e.g. if the light is on and one switch is off, the other switch must be on, etc.). The second example is from Judea Pearl (who is the main modern theorist in this domain). It concerns a car that will not start. This may be because of a dead battery, or it may be because the car has no gas.

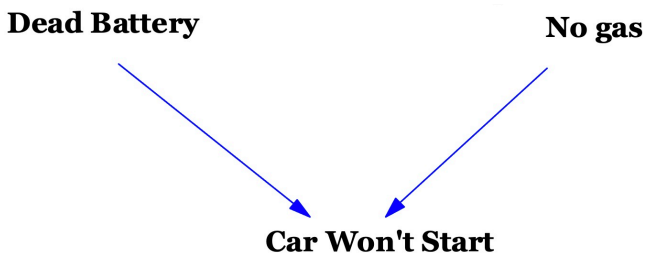


Figure 47. A Collider Between Possible Causes of a Car Not Starting.

Dead Battery is entirely independent of No gas. But in the path **Dead Battery -> Car Won't Start <- No gas**, if we conditionalize on the variable **Car Won't Start** then **Dead Battery** becomes conditionally dependent on **No gas**. For example, if you are told that the car won't start (i.e. that the value for the **Car Won't Start** variable is True) then knowing that the battery is good tells you that the car does not have any gas. In sum here, where there is a common effect, a collider, there is no correlation (or dependence) between the causes. But if the analysis conditions on the common effect there is conditional correlation (or conditional dependence) between the causes. This is called *collider bias*. It is worth mentioning because more than a few times in real research publications *it happens by accident*. Julia Rohrer mentions the following example. Say we are interested in whether rigorous research is correlated with innovative research— no causality being looked for here, just association— and we decide, being fancy, to look at this question with published research being considered separately from unpublished research. But rigorous research causes it to be published, and innovative research causes it to be published, so being published is a collider. Our conditioning on the collider may produce association out of thin air— collider bias (Rohrer 2018, 35).

Let us now briefly revisit the ordinary chain structure with one or more link or mediating variables. Say:

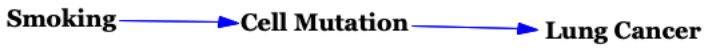


Figure 48. A Chain From Smoking to Lung Cancer Mediated by Cell Mutation.

As it stands there will be three dependencies or associations or correlations: **Smoking: Cell Mutation, Cell Mutation: Lung Cancer, Smoking: Lung Cancer**. But consider what happens were we to conditionalize on the link variable, i.e. on Cell Mutation. Knowledge of a value for the cell mutation so-to-speak masks any values for the smoking variable as far as predicting the lung cancer in concerned. Suppose the variable **Cell Mutation** can have just two values: that there is mutation and that there is no mutation. We know in general that smoking is associated with lung cancer. This means that in some cases varying whether there is smoking or not brings about whether there is lung cancer or not. But is smoking associated with lung cancer when there definitely is cell mutation? The answer is 'No'. Varying the smoking has no effect on the cell mutation and it is the cell mutation that is the direct cause of the lung cancer. It is similar for the other case where we fix the link value as being no mutation. In sum here, where there are link variables in a chain, there is correlation (or dependence) between all the upstream variables of a chosen link, and the link, and all the downstream variables of the link. But if the analysis

conditions on the chosen link there is no conditional correlation (or conditional dependence) between all the upstream variables of a chosen link, and the link, and all the downstream variables of the link.

The causal diagrams can be much more complicated than those displayed here. Minimally there are diagrams like this:

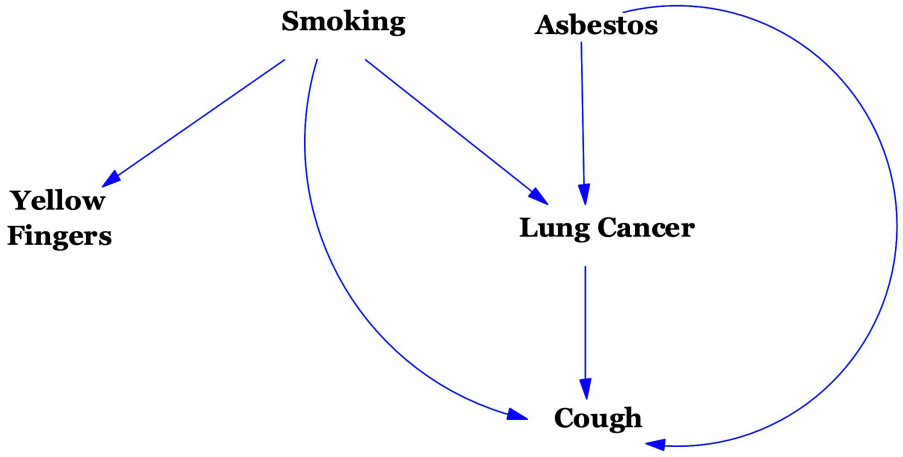


Figure 49. A Complex Interplay of Causes Around Smoking.

Lung cancer causes coughs. Asbestos causes lung cancer and, without causing lung cancer, causes coughs. Smoking also causes coughs without causing lung cancer. Smoking causes yellow fingers.

The diagrams use arrows and these give a direction or flow to time and causality— which variables cause which other variables and which variables are 'earlier' than others. This brings another consideration into focus. No variable can cause itself. This means that no directed path in a causal diagram (i.e. a path following the direction of the arrows) can go around in

a circle or 'cycle' and come back to an earlier variable. This means that the diagram, the graph structure as a whole, is a Directed Acyclic Graph (DAG).

Let us review the presentation. Causal diagrams can have two meanings or functions: a data free explanation or description of the causes that are presumed to be at work, and a potentially data rich statistical description of the associations or dependencies or correlations. The causality has a direction (if smoking is the cause of lung cancer, lung cancer is not the cause of smoking). The dependencies do not have directions (if smoking is associated with lung cancer, lung cancer is associated with smoking). Variables can be independent of each other. There is also the notion of conditional dependence and independence. The various structures of the graphs (the chains, forks, and collisions) give rise to the various dependencies. They give rise to complex statistical predictions that can be tested. For example, the very simple causal diagram:

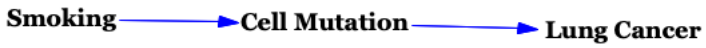


Figure 50. A Chain From Smoking to Lung Cancer Mediated by Cell Mutation.

entails that smoking is correlated with lung cancer and not conditionally correlated with lung cancer when cell mutation is controlled for. If either of these statistical prediction are mistaken, so too is the original causal diagram.

There is a technique, in fact an algorithm, *d-separation* ('direction separation') that can convert a causal diagram into all the statistical predictions that it entails.

To sum up this whole presentation in a simple way. If there are forks, there needs to be conditionalization on the common causes. If there are collisions, there needs to be *no* conditionalization on the colliders.

The importance of the causal diagrams is this. Almost all, maybe even all, machine learning is about what causes what— about learning about causality. Machine learning, and all other kinds of empirical science for that matter, never has anything more to work with than correlations. They all have to make the leap from correlation to causation. There is no way this can be done infallibly. There is no way it can be done without assumptions. But causal diagrams provide a framework for making assumptions and for suggesting the appropriate correlations to test.

Let us conclude by sketching two examples of how causal diagrams might be used in connection with bias. Matt Kusner et al. introduce the example of predicting the First Year Average Grade (FYA) of students in law school (Kusner et al. 2018). They write:

> The Law School Admission Council conducted a survey across 163 law schools in the United States [35]. It contains information on 21,790 law students such as their entrance exam scores (LSAT), their grade-point average (GPA) collected prior to law school, and their first year average grade (FYA).

> Given this data, a school may wish to predict if an applicant will have a high FYA. The school would also like to make sure these predictions are not biased by an individual's race and sex. However, the LSAT, GPA, and FYA scores, may be biased due to social factors. (Kusner et al. 2018)

A causal diagram for this might be:

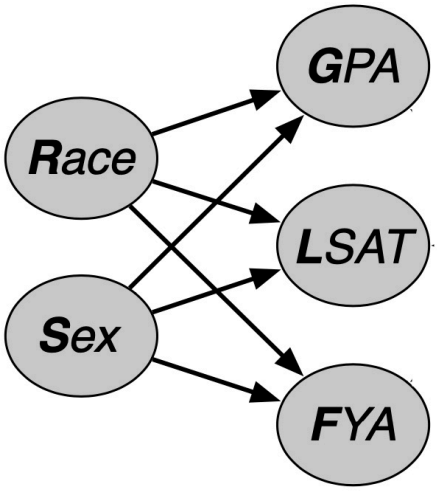


Figure 51. Law School First Year Average Grade (Kusner et al. 2018).

This asserts that Race non-deterministically affects GPA, LSAT, and FYA, as does Sex. In this problem, it is imagined that the Law Schools want to predict first year average grade FYA and they cannot use Race and Sex as these are protected. The next step is to realize, or take into account, that knowledge, a student's knowledge, also affects GPA, LSAT, and FYA, and this gives a second causal diagram.

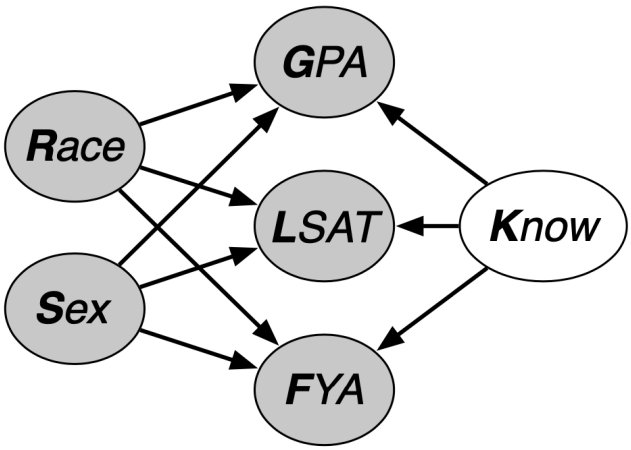


Figure 52. Law School First Year Average Grade With Student Knowledge (Kusner et al. 2018).

With this causal diagram, some mathematics can be done on the forks, collisions, and the data, that can produce the requisite predictions of FYA from LSAT and GPA without using the protected attributes of Race and Sex. The Law Schools would be able to prove the predicted FYA is not biased.

Typically, the diagrams are more complex. This one from Tyler VanderWeele and Nancy Staudt relates legal cases and their case characteristics to both judicial decisions and the likelihood that litigation will take place in courts with judges of a particular race, gender, age or ideology (for plaintiffs prefer to file claims with judges deemed friendly to their legal claims) (VanderWeele and Staudt 2011).

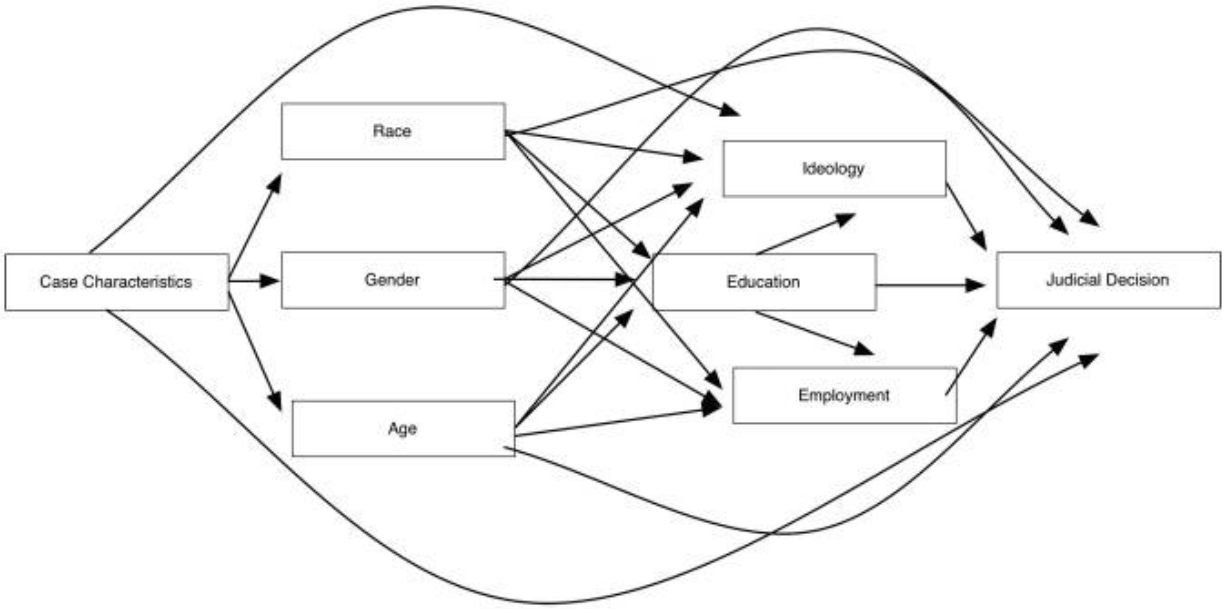


Figure 53. A Depiction of Possible Factors in a Judicial Decision (VanderWeele and Staudt 2011).

In sum. If there is an assertion of bias, bias in allocation, say, perhaps with allocation of mortgages or allocation of places in a Law School. Suitable causal diagrams might be able to highlight relevant correlations to be investigated. In turn, these might be able to produce evidence, within the bounds of fallibility, that there is no bias. The diagrams provide a structure for investigation false positives and false negatives. Separately, explainable artificial intelligence (XAI) is important. Explanations need causes. Causal diagrams are a steppingstone, along with data and correlations, towards identifying causes.

D.3 Annotated Readings for Appendix D

Hernan, Miguel. “Causal Diagrams: Draw Your Assumptions Before Your Conclusions.” edX, 2022. https://www.edx.org/course/causal-diagrams-draw-your-assumptions-before-your. (Hernan 2022)

Myint, Leslie. “Key Structures in Causal Graphs”, 2020. https://www.youtube.com/watch?v=UA0vyBnzi9U. (Myint 2020) Leslie Myint has posted many excellent videos on statistics and causality. This is a relevant example.

Rohrer, Julia M. “Thinking Clearly About Correlations and Causation: Graphical Causal Models for Observational Data.” *Advances in Methods and Practices in Psychological Science* 1, no. 1 (2018): 27–42. https://doi.org/10.1177/2515245917745629. (Rohrer 2018). This is a masterpiece. It is probably a little advanced for us.

Appendix E: Knowledge Graphs

E.1 Knowledge Graphs

A knowledge graph is a means of representing knowledge. Its relevance to us is that is a technology or technique that can help with the search and discovery of information. It is not in itself an artificial intelligence or machine learning technique. However, it can be part of an infrastructure that supports some applications of machine learning in librarianship.

Conceptually, a knowledge graph starts with a node representing an object, which might be a person, a place, or a thing, or a date, or etc., then adds as links facts about that object. Here is a very simple knowledge graph:

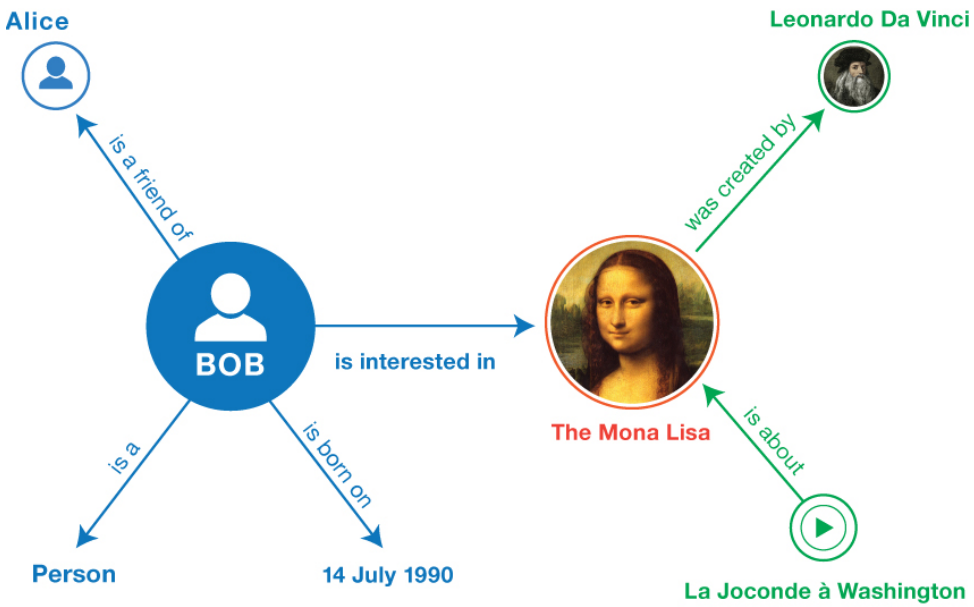


Figure 54. A Simple Knowledge Graph (W3C Working Group 2014).

Typical knowledge graphs will be much more complicated than this. Google have a knowledge graph technology, 'Google Knowledge Graph'. As of 2012 this had 500 million objects and 3.4 billion facts in it (Singhal 2012). Google populate their Knowledge Graph using search data that they have from users. For example, if the search 'who painted the Mona Lisa?' was popular, the answer to that search would be added to the knowledge graph as a fact.

Usually, a knowledge graph is a data structure that computer programs can use in the background (as opposed to the user exploring it visually). No user is going to look through a graph of 500 million objects.

Some possibilities for knowledge graphs are:

1. Helping disambiguate. For example, were there to be several 'Bob' objects in the graph, the user or system would know that disambiguation was required for queries about 'Bob' and maybe suggest facts to do it.
2. Helping with creating summaries of information about an object. For example, it could help identify key facts (say about the Mona Lisa).
3. Pointing the user to further popular (or rare) queries or information about the objects.
4. Being part of a question answering system (for example, who painted Mona Lisa?)

5. Being part of a recommender system (for example, by extracting the preferences of similar people in the graph).

E.2 Annotated Readings for Appendix E

Peng, Ciyuan, Feng Xia, Mehdi Naseriparsa, and Francesco Osborne. “Knowledge Graphs: Opportunities and Challenges.” *Artificial Intelligence Review* 56, no. 11 (2023): 13071–102. https://doi.org/10.1007/s10462-023-10465-9.(Peng et al. 2023).

Glossary

There are many excellent ML glossaries available online, for example Google's *Machine Learning Glossary: ML Fundamentals* (Google for Developers 2023) (We are using and editing parts of that here.) There are also 'explanations of key concepts', approaching them from a librarianship point of view, for example, Brady Lund and Ting Wang's *Chatting about ChatGPT: how may AI and GPT impact academia and libraries?* (Lund and Wang 2023).

Generally, we are restricting ourselves to technical or unusual terms that appear in this text.

A

accuracy

The number of correct classification predictions divided by the total number of predictions.

application programming interfaces (APIs)

the protocols that allow outside programs to use, or communicate with, an application.

algorithm

1. a step-by-step computational procedure.
2. any piece of computer software, especially software that can make decisions independent of a human decision maker.

alignment

that the model's predictions or behavior correspond closely with the expected or desired or intended outcome.

anaphora

the use of pronouns or other words to refer back to previously mentioned subjects or objects.

artificial intelligence (AI)

the use of computers, algorithms, and sometimes outside data, to solve problems that an ideally rational and intelligent human being would be able to solve, given the time, resources, and ingenuity.

artificial general intelligence (AGI)

a non-human mechanism that demonstrates a *broad range* of problem solving, creativity, and adaptability.

artificial super intelligence (ASI)

is the possibility that AGI becomes established and then the AGI machines simply design themselves and become smarter and smarter in a runaway fashion.

attention

a technique that enables models to dynamically focus on certain parts of the input for better performance, especially in sequence-to-sequence tasks.

B

bag of words

a representation of the words in a phrase or passage, irrespective of order.

bias (ethics/fairness)

1. unfairness by means of stereotyping, prejudice or favoritism towards some things, people, or groups over others.
2. errors in input data or output predictions (independent of questions of fairness or unfairness).

C

cause (or causal factor)

a) X and Y occur, b) X precedes Y in time, c) the presence of X raises the probability of Y, and d) had there been an intervention that changed X, that would change the probability of Y. For example, the diagnosis that smoking caused lung cancer in a group of patients amount to a) the patients smoked and have lung cancer, b) the patients smoked before they got lung cancer, c) their smoking raised their probability of getting lung cancer, and d) had they not smoked in the first place, their probability of getting lung cancer would have been lower.

chain-of-thought (COT) prompting

encouraging a large language model (LLM) to explain its reasoning, step by step. For example, consider the following prompt, paying particular attention to the second sentence:

> How many g forces would a driver experience in a car that goes from 0 to 60 miles per hour in 7 seconds? *In the answer, show all relevant calculations.*

Chain-of-thought prompting forces the LLM to perform all the calculations, which might increase the chances of the answer being correct. In addition, chain-of-thought prompting enables the user to examine the LLM's steps to determine whether the answer makes sense.

chatbot

a computer program designed to simulate conversation with human users, especially over the internet.

class

a labeled collection of items. For example:

1. In a model that detects spam, the two classes might be spam and not-spam.
2. In a model that identifies dog breeds, some of the classes might be poodle, beagle, pug.

classification model

predicts a class. For example, the following are all classification models:

1. A model that predicts an input sentence's language (French? Spanish? Italian?).
2. A model that predicts tree species (Maple? Oak? Baobab?).
3. A model that predicts the positive or negative class for a particular medical condition.

cloze task

a method used in language teaching and as a machine learning task. Such tasks are often referred to as a fill-in-the-blanks task, cloze test, gap-filling task, or text completion task. It involves having a model fill in missing words or tokens in a sentence or paragraph.

clustering

grouping related examples, particularly during unsupervised learning. Once all the examples are grouped, a human can optionally supply a label or meaning to each cluster for example that cluster 1 is ‘dwarf trees’ and cluster 2 is ‘full-size’ trees.

conversational implicature

the meaning that is implied by a speaker but not explicitly stated in the conversational context.

conjectural

speculative, hypothetical, theoretical, presumptive, or fallible. There is a philosophical view, highlighted by Karl Popper, that all knowledge of the world, and some knowledge of mathematics and logic, are conjectural.

confirmation bias

the tendency to search for, interpret, favor, and recall information in a way that confirms one's preexisting beliefs or hypotheses.

controlled vocabulary

a standardized terminology, curated term set, or fixed lexicon. It is a predefined set of terms that are used to ensure consistency in the tagging and categorization of content.

constraint prompting

a method to direct a model's generation or response by setting specific constraints or conditions or by providing a template that should be used for the answer.

counterfactual

Let us start with an example. Napoleon lost the Battle of Waterloo, as a matter of fact. But an adventurous historian might consider the question of what would have happened if Napoleon had won the Battle of Waterloo. Supposing that Napoleon won would be counter-to-the-facts and the historian’s analysis would be a counterfactual analysis.

Thinking counterfactually, which we do all the time, especially when making decisions, involves considering alternative scenarios and outcomes that did not happen or might not have happened but which could have happened under different circumstances. (If I'd run faster, I would have caught the bus. If I'd studied harder, I would have passed the exam. If inflation had continued, there would have been a fall in unemployment. etc.) Some counterfactuals are true, and some are false (for example, if the aforementioned bus were travelling at 50mph no amount of faster running on my part would result in my catching it). Counterfactual reasoning and causal reasoning are often intertwined. Were we to say 'John Smith's lung cancer was caused by smoking' we in part mean 'If John Smith had not smoked, he would not have got lung cancer' (which is counterfactual). Then, were we to say 'If Jane Smith, a non-smoker, were to become a smoker, she would raise the probability of her getting lung cancer', that statement is a true counterfactual supported by a causal connection between smoking and lung cancer. Overall, counterfactuals are a valuable tool in many fields for analyzing and understanding the implications of events and decisions by considering alternatives that did not occur.

CPUs, GPUs, and CUDA®

a CPU is a central processing unit, the main computing component of a standard computer. A GPU is a graphics processing unit, a computing component originally designed to provide accelerated graphics for video games and similar. Typically, a CPU works serially, one task after another, whereas a GPU works in parallel, carrying out many different tasks at the same time. The company NVIDIA realized that GPUs, which they specialized in, were ideal for artificial intelligence. They produced the

CUDA® platform, which is for high performance, high throughput, computing (not necessarily having anything to do with graphics). As of 2024, most LLM research and commercial work will be using the CUDA® platform and NVIDIA electronic chips.

D

data set or dataset

raw data, commonly organized in one of the following formats:

1. a spreadsheet
2. a file in CSV (comma-separated values) format

deep learning (DL)

advanced techniques within the field of artificial intelligence focused on mimicking the operation of the human brain.

deep model

neural networks containing more than one hidden layer.

deepfakes

media that have been altered, or wholly generated, by artificial intelligence to present something that did not actually occur. Examples include manipulated videos which might include realistic speeches from politicians that simply never happened.

delayed rewards

in many settings there are rewards (such as passing the exam, arriving at the destination safely, finding the cheese in the maze.) Delayed rewards are where the agent has to do some exploring of the task or making more than a few steps presumably towards the goal, before a reward is given. The above examples have delayed rewards— pressing a button to get a pellet of food does not.

digitization

the conversion of text, pictures, or sound into a digital form that can be processed by a computer.

discriminative model

predicts the appropriate or correct labels for new examples presented to it. For example, a discriminative model might predict whether incoming email was spam. It discriminates between spam and not-spam. In contrast, generative models might produce or create completely new examples such as new images or paintings. Most supervised learning models are discriminative models.

dynamic

The terms dynamic and online are synonyms in machine learning. The following are common uses of dynamic and online in machine learning:

1. A dynamic model (or online model) is a model that is retrained frequently or continuously. A dynamic model is a 'lifelong learner' that constantly adapts to evolving data.
2. Dynamic training (or online training) is the process of training frequently or continuously.
3. Dynamic inference (or online inference) is the process of generating predictions on demand.

E

embeddings

numerical representations of text, concepts, or other types of data as lists of numbers (i.e. they are points or vectors in a high-dimensional space).

empirical

knowledge or models that are based on observation or experience rather than on pure mathematics or pure logic. Empirical knowledge is tested by means of experiments or observations.

epoch

a full training pass over the entire training set such that each example has been processed once.

example

The values of one row of features and possibly a label. Examples in supervised learning fall into two general categories:

1. A labeled example consists of one or more features and a label. Labeled examples are used during training.
2. An unlabeled example consists of one or more features but no label. Unlabeled examples are used during inference and use.

For instance, a model is being trained to determine the influence of weather conditions on student test scores. Here are three labeled examples:

Features			**Label**
Temperature	**Humidity**	**Pressure**	**Test score**
15	47	998	Good
19	34	1020	Excellent
18	92	1012	Poor

Figure 90. Labeled Examples.

Here are three unlabeled examples:

Features		
Temperature	**Humidity**	**Pressure**
12	62	1014
21	47	1017
19	41	1021

Figure 91. Unlabeled Examples.

In supervised machine learning, models train on labeled examples and make predictions on unlabeled examples.

In semi-supervised and unsupervised learning, unlabeled examples are used during training.

expert system

a computer system that emulates the decision-making ability of a human expert.

F

fakes

in machine learning, fakes are the products of algorithms designed to generate deceptive content that mimics the real one.

fallible

not absolutely certain. Many views, theories, and observation reports are true. Nevertheless, often it is not known with absolute certainty whether they are true. Our knowledge of them is fallible.

false negative

an example in which the model mistakenly predicts a member of the negative class. For example, the model predicts that a particular email message is not spam (the negative class), but that email message actually is spam.

false positive

an example in which the model mistakenly predicts a member of the positive class. For example, the model predicts that a particular email message is spam (the positive class), but that email message is actually not spam.

feature

An input variable to a machine learning model. An example consists of one or more features. For instance, suppose a model is being trained to determine the influence of weather conditions on student test scores. The following table shows three examples, each of which contains three features and one label:

Features			**Label**
Temperature	**Humidity**	**Pressure**	**Test score**
15	47	998	Good
19	34	1020	Excellent
18	92	1012	Poor

Figure 92. Features With Labels.

few-shot learner

a machine learning approach where the, perhaps pre-trained, model is designed to learn information with a small amount of, perhaps additional, training data.

few-shot prompting

prompting where a few examples are given of what the LLM should do.

fine tuning

taking a pre-trained model and adjusting its parameters slightly to adapt to a new but related task.

foundation models

starting points for creating more specialized machine learning models.

frame

a frame, within machine learning and artificial intelligence, is a data structure for representing a stereotypical situation, like a room and its contents, ordering a meal in a restaurant, or buying an airline ticket.

G

generative

the capability of some models to generate completely new data instances that resemble the training data— for example new images or paintings, or new poems or essays.

genial understander system

a natural language understanding system which can interact with users friendly or amiable manner to find out what it is they want and any requisite data to assist in that task (e.g. to discover data for a frame or frames).

generalization

making correct predictions on new, previously unseen data. A model that can generalize is the opposite of a model that is overfitting.

GPUs, CPUs, and CUDA®

See CPUs, GPUs, and CUDA®.

gradient descent

a mathematical technique used in training to minimize loss or error. Gradient descent iteratively adjusts parameters of the model, gradually finding the best combination to minimize loss.

ground truth

Reality. For example, consider a model that predicts whether a student in their first year of university will graduate within six years. Ground truth for this model is whether that student actually graduated within six years.

H

hallucinations

false generations, spurious outputs, or fictitious predictions. It is when a model generates information that is ungrounded or not supported by the input data.

homograph

a word characterized by lexical ambiguity, as it shares the same spelling as another word but has a different meaning for example ‘bank’ (of river) and ‘bank’ (financial).

I

inductive bias

training data, no matter how extensive, is never enough on its own to determine predictions about new unseen data. Assumptions are needed. Those assumptions are inductive bias.

inference

the process of making predictions by applying a trained model to unlabeled examples.

interpretability

explaining or to presenting a model's reasoning in terms understandable to a human.

intersubjective (shared conventional knowledge)

conventional agreements shared by multiple individuals or systems. For example, a library classification system such as the Dewey Decimal System is a shared conventional agreement.

K

keyword search

retrieving information by matching query terms ('keywords') against a set of documents or a database.

L

label

in supervised machine learning, the 'answer' or 'result' portion of an example. Each labeled example consists of one or more features and a label. For instance, in a spam detection dataset, the label would probably be either 'spam' or 'not spam.' In a rainfall dataset, the label might be the amount of rain that fell during a certain period.

labeled example

See example.

language model

a statistical model, a generative model, or a predictive model of language. It is a type of model that can predict the next word in a sentence or help generate text based on previous text.

large language models (LLMs)

usually generative pre-trained transformers. These are highly complex models designed to understand, generate, and translate human language.

layer

A set of neurons in a neural network. Three common types of layers are as follows:

1. The input layer, which provides values for all the features.
2. One or more hidden layers, which find relationships between the features and the label.
3. The output layer, which provides the prediction.
4.

For example, the following illustration shows a neural network with one input layer, two hidden layers, and one output layer:

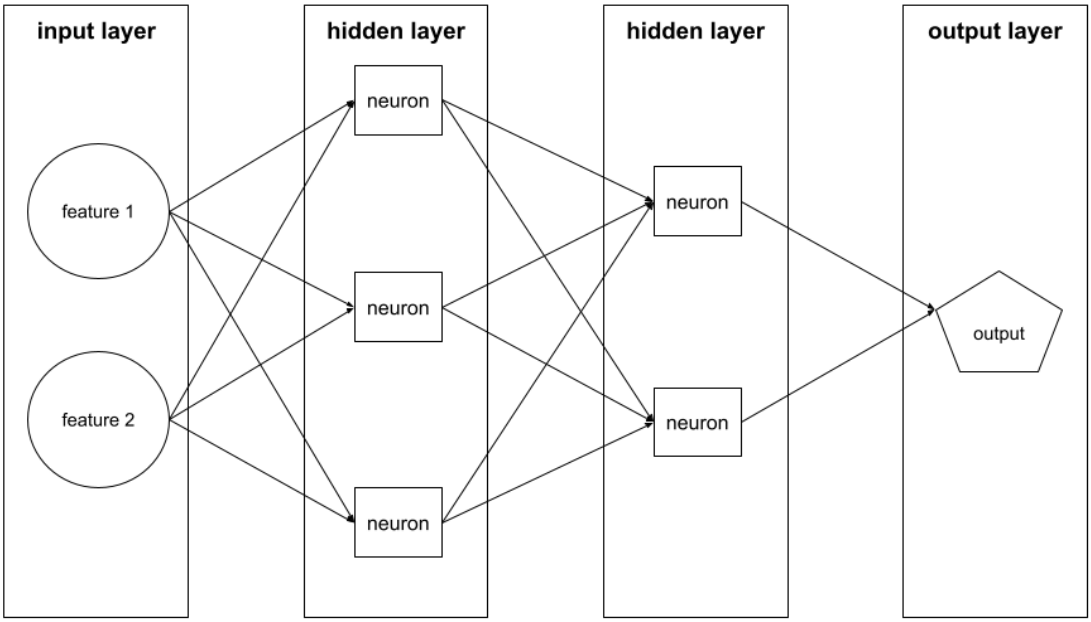


Figure 93. A Neural Network.

M

machine learning (ML)

a program or system that trains a model from input data. The trained model can make useful predictions from new (never-before-seen) data drawn from the same distribution as the one used to train the model.

Markov (process or chain)

a mathematical system that undergoes transitions from one state to another, with probabilistic rules that depend only on the current state and not on the sequence of events that preceded it. It is a memoryless model. It 'knows' the state that it is in, but 'knows' nothing of the states before that. The child's game Snakes and Ladders is an example of a Markov process.

modality

a high-level data category. For example, numbers, text, images, video, and audio are five different modalities.

model

any mathematical construct that processes input data and returns output. Phrased differently, a model is the set of parameters and structure needed for a system to make predictions. In supervised machine learning, a model is trained on labeled examples, then, in use, it takes an unlabeled example as input and infers a prediction of the correct label as output. Unsupervised machine learning also generates models, typically a function that can map an input example to the most appropriate cluster for it.

N

negative class

in binary classification, one class is termed positive and the other is termed negative. The positive class is the thing or event that the model is testing for and the negative class is the other possibility. For example:

1. The negative class in a medical test might be 'not tumor.'
2. The negative class in an email classifier might be 'not spam.'

neural network

a model containing at least one hidden layer. A deep neural network is a type of neural network containing more than one hidden layer. For

example, the following diagram shows a deep neural network containing two hidden layers.

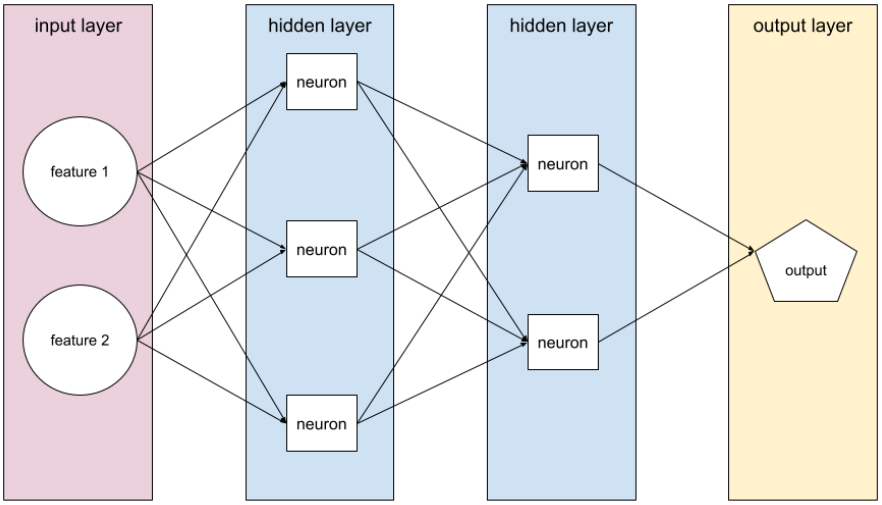


Figure 94. A Neural Network Showing Hidden Layers.

Each neuron in a neural network connects to all of the nodes in the next layer. For example, in the preceding diagram, notice that each of the three neurons in the first hidden layer separately connect to both of the two neurons in the second hidden layer.

Neural networks implemented on computers are sometimes called artificial neural networks to differentiate them from neural networks found in brains and other biological nervous systems.

Some neural networks can mimic extremely complex relationships between different features and the label.

neuron

In machine learning, a neuron is a distinct unit within a hidden layer of a neural network. Each neuron perform does a calculation on its own inputs and decides whether it is activated i.e. whether it fires or is 'triggered'.

A neuron in the first hidden layer accepts inputs from the feature values in the input layer. A neuron in any hidden layer beyond the first accepts inputs from the neurons in the preceding hidden layer. For example, a neuron in the second hidden layer accepts inputs from the neurons in the first hidden layer.

The following illustration highlights two neurons and their inputs.

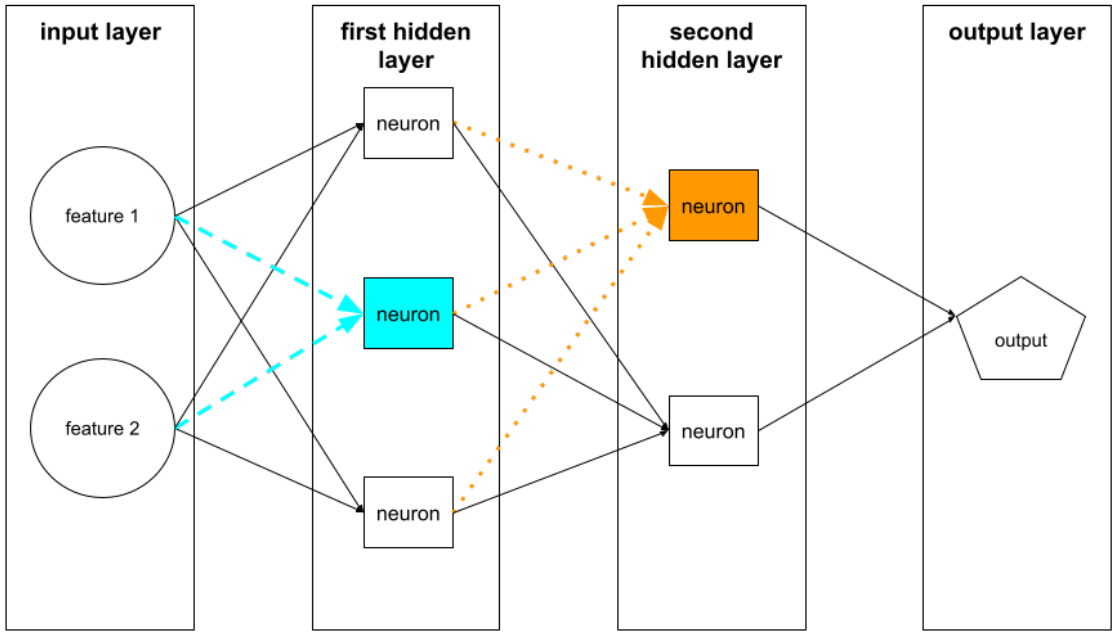


Figure 96. A Neural Network Showing Some Neurons.

A neuron in a neural network mimics the behavior of neurons in brains and other parts of nervous systems.

node (neural network)

A neuron in a hidden layer.

O

objective knowledge

factual knowledge and the knowledge of science and mathematics.

offline

synonym for static.

one-shot prompting

prompting where a single example is given.

online

synonym for dynamic.

ontology

a conceptual framework, a knowledge representation schema, a semantic framework, or a taxonomy, particularly if it is structured hierarchically.

online inference

generating predictions on demand. For example, suppose an app passes input to a model and issues a request for a prediction. A system using

online inference responds to the request by running the model (and returning the prediction to the app).

optical character recognition (OCR)

a technology for text recognition, character reader technology, document digitization, and image-to-text conversion. It converts different types of documents, such as scanned paper documents, PDF files, or images captured by a digital camera, into editable and searchable data.

output layer

The 'final' layer of a neural network. The output layer contains the prediction.

The following illustration shows a small deep neural network with an input layer, two hidden layers, and an output layer:

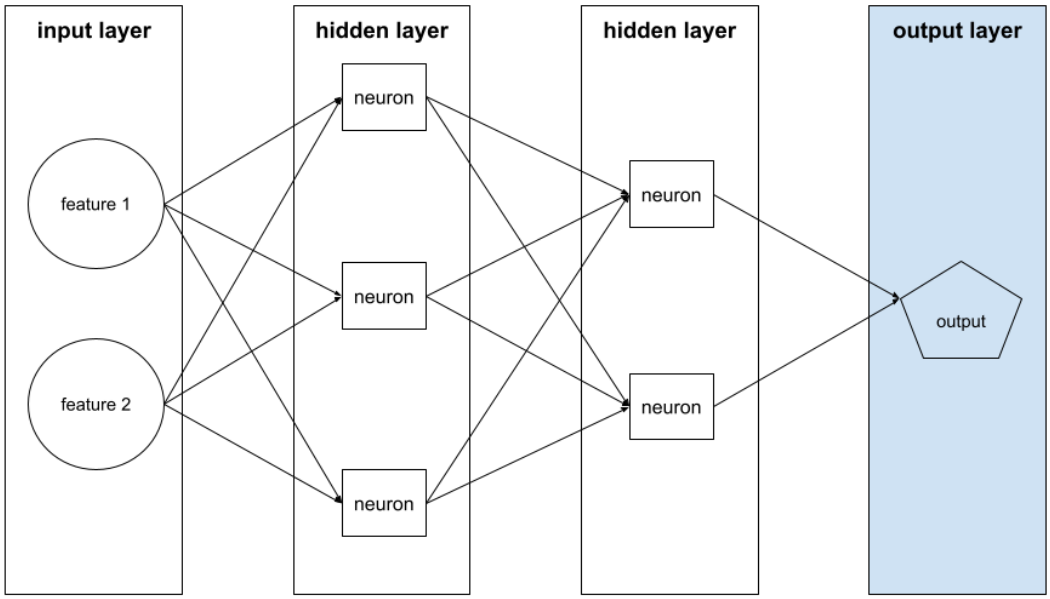


Figure 96. A Neural Network Showing An Output Layer.

overfitting

creating a model that matches the training data so closely that the model fails to make correct predictions on new data that is not part of the training.

P

panopticon

a prison where all actions of the prisoners are observable by the controlling entity.

parameter

internal variables that can be adjusted during training to alter the model's behavior. Suitable, or the best, values for the parameters are learned during training.

paternalism

to act for the good of another person without that person's consent, as parents do for children (Suber 1999).

positive class

the class that is being testing for. For example, the positive class in a cancer model might be 'tumor.' The positive class in an email classifier might be 'spam.'

post-processing

adjusting the output of a model after the model has been run. Post-processing can be used to enforce fairness constraints without modifying models themselves. For example, one might apply post-processing to a binary classifier by setting a classification threshold such that equality of opportunity is maintained for some attribute by checking that the true positive rate is the same for all values of that attribute. As an example of this, in a mortgage application program, post-processing might check that those living north of the railway tracks get as many mortgages as those living south of the railway tracks.

precision

Consider a search of a library collection for items that are relevant to that search. Precision is the proportion, or percentage, *of the returned items* that are relevant. So, if everything returned is relevant, precision is 100%. If half the items returned are relevant, precision is 50%. Precision has a companion property 'recall'. Recall is the proportion, or percentage, *of the relevant items in the collection* that are returned. If all the relevant items are returned, recall is 100%. If half the relevant items in the collection are returned, recall is 50%.

Machine learning uses similar concepts of precision and recall. Precision is the proportion of true positives among the positives in a classification task. If every positive is a true positive, precision is 100%. If half the positives are true positives (and half are false positives) precision is 50%. If there are no false negatives (i.e. all the positives are captured), recall is 100%. If half of

the positives get classified as negatives (i.e. half of the positives get misclassified as (false) negatives), recall is 50%.

prediction

a model's output. For example:

1. The prediction of a binary classification model is either the positive class or the negative class.
2. The prediction of a multi-class classification model is one class.
3. The prediction of some models can be a number.

Q

quasi-empirical

investigative approaches or knowledge that are similar to those of empirical research. That is, they might use observation and experiment in discovery and testing. For example, in mainstream computer science, determining what typical algorithms will do when run is a matter of mathematical proof. There is no need to run the programs or observe them running. Such knowledge is non-empirical. But, in machine learning, often finding out what the software will do is a matter of trying it, observing its behavior, and even conducting experiments. In such areas, the knowledge is quasi-empirical.

R

rater

a human who provides labels for examples. ‘Annotator’ is another name for rater.

recall

Consider a search of a library collection for items that are relevant to that search. Recall is the proportion, or percentage, *of the relevant items in the collection* that are returned. If all the relevant items are returned, recall is 100%. If half the relevant items in the collection are returned, recall is 50%. Recall has a companion property ‘precision’. Precision is the proportion, or percentage, *of the returned items* that are relevant. So, if everything returned is relevant, precision is 100%. If half the items returned are relevant, precision is 50%.

Machine learning uses similar concepts of recall and precision. If there are no false negatives (i.e. all the positives are captured) in a classification task, recall is 100%. If half of the positives get classified as negatives (i.e. half of the positives get mis-classified as (false) negatives), recall is 50%. Precision is the proportion of true positives among the positives. If every positive is a true positive, precision is 100%. If half the positives are true positives (and half are false positives) precision is 50%.

recommender system

a system that predicts and provides suggestions tailored to the user's preferences.

reinforcement learning

a learning paradigm that trains algorithms based on a system of rewards and penalties.

reliability

the degree to which a measurement, prediction, or algorithm yields stable consistent results over multiple runs or data sets. A bathroom scale that always weighs 5 pounds light is reliable (but it is not accurate and nor are its results valid).

retrieval-augmented generation (RAG)

is a technique to both to keep an LLM up to date with what it 'knows' and to be more accurate in its replies. The idea is to give the LLM access to an external database or databases. Then factual prompt questions to the LLM are augmented with the instruction to check with the databases and find supporting facts, references, and citations. As external knowledge grows there is no need to re-train the LLM. Rather, all that is required is for the databases to be updated (which they usually would be as a matter of course, say for news articles).

S

self-supervised

a machine learning technique that involves algorithms that can learn to label, classify, or predict new instances without explicit human-provided

labels. Once the labels exist, the technique can use standard supervised learning.

semantic search

meaning-based search, conceptual search, or context-aware retrieval. It uses search algorithms that understand the searcher's intent and the contextual meaning of terms to fetch more relevant results.

sentiment analysis

computationally determining and categorizing opinions expressed in a piece of text, especially to determine the writer's attitude towards a particular topic.

specification (for a computer program)

the program design, software requirements, or system design. It outlines the expected functions, behaviors, and structures of the computer program.

static

done once rather than continuously. The terms static and offline are synonyms. The following are common uses of static and offline in machine learning:

1. static model (or offline model) is a model trained once and then used for a while.
2. static training (or offline training) is the process of training a static model.

3. static inference (or offline inference) is a process in which a model generates a batch of predictions at a time.

static inference

synonym for offline inference.

stochastic (processes)

random, non-deterministic, or probabilistic processes.

stochastic psittacosis

[this is a joke.] Emily Bender describes large language models as being stochastic parrots. Psittacosis is a disease that parrots can have. Hence stochastic psittacosis captures the shortcomings of large language models.

supervised machine learning

training a model from features and their corresponding labels. Supervised machine learning is analogous to learning a subject by studying a set of questions and their corresponding answers. After mastering the mapping between questions and answers, a student can then provide answers to new (never-before-seen) questions on the same topic.

switch transformer

a type of transformer model designed to handle extremely large-scale datasets and models by activating only a portion of the model at a time.

synonym (pair)

a pair of equivalent terms, lexical equivalents, alternative terms, or interchangeable words. These are words or phrases that have the same or nearly the same meanings and can be used to provide variety in text or speech, for example ‘attorney’ and ‘lawyer’.

T

theory-laden

observations, especially those using instruments, can be said to be theory-laden and this means that they are influenced by underlying theoretical frameworks. For example, measuring the temperature of a medical patient using an ordinary glass thermometer takes for granted background theories about the expansion of mercury and glass. That observations are theory-laden does not mean that any or all of them are false or incorrect. It does, however, mean that they are fallible (i.e. caution is needed, they might be mistaken).

token

is used in two different ways. Were we to ask, how many letters are there printed below?

t t t

The answer ‘one’ would be talking of the letter tee as a *type*. Whereas the answer ‘three’ is talking of the *tokens* of the letter tee. The second sense of

‘token’ concerns small processing units. For example, LLMs have a context window of text. That text consists of tokens, the chunks of text that the LLM looks at. Typically, tokens are larger than individually characters but smaller than whole words. So, a context window of 1000 tokens might amount to a context window of 1000 words.

training

the process of determining the ideal parameters comprising a model. During training, a system reads in examples and gradually adjusts parameters. Training uses each example anywhere from a few times to billions of times.

training set

The subset of the dataset used to train a model.
Traditionally, examples in the dataset are divided into the following three distinct subsets:

1. a training set
2. a validation set
3. a test set

Ideally, each example in the dataset should belong to only one of the preceding subsets. For example, a single example should not belong to both the training set and the validation set.

transformers

a type of model in machine learning known as self-attention models, transformer architectures, sequence-to-sequence models, or attention-based models. These are particularly powerful in handling sequences of data, such as natural language, for tasks like translation or summarization.

true negative

an example correctly predicted by the model as belonging to the negative class. For example, the model infers that a particular email message is not spam, and that email message really is not spam.

true positive

an example correctly predicted by the model as belonging to the positive class. For example, the model infers that a particular email message is spam, and that email message really is spam.

U

underfitting

a model with poor predictive ability caused by the model not having fully captured the complexity of the training data. Many problems can cause underfitting, including:

1. Training on the wrong set of features.
2. Training for too few epochs.
3. Providing too few hidden layers in a deep neural network.

unfairness

synonym for one meaning of bias.

unlabeled example

see example.

unsupervised machine learning

training a model to find patterns in a dataset, typically an unlabeled dataset. The most common use of unsupervised machine learning is to cluster data into groups of similar examples. For example, an unsupervised machine learning algorithm can cluster songs based on various properties of the music. The resulting clusters can become an input to other machine learning algorithms (for example, to a music recommendation service). Clustering can help when useful labels are scarce or absent. For example, in domains such as anti-abuse and fraud, clusters can help humans better understand the data.

V

validation

the initial evaluation of a model's quality. Validation checks the quality of a model's predictions against the validation set. Because the validation set differs from the training set, validation helps guard against overfitting. Evaluating the model against the validation set can be thought of as the

first round of testing and evaluating the model against the test set as the second round of testing.

validation set

the subset of the dataset that performs initial evaluation against a trained model. Typically, you evaluate the trained model against the validation set several times before evaluating the model against the test set.

Traditionally, you divide the examples in the dataset into the following three distinct subsets:

1. a training set
2. a validation set
3. a test set

Ideally, each example in the dataset should belong to only one of the preceding subsets. For example, a single example should not belong to both the training set and the validation set.

validity

accuracy, correctness, or soundness. It concerns the extent to which a model or method accurately measures or predicts what it is intended to.

Bibliography

A12 Allen Institute for AI. 2022. “About — ARC: AI2 Reasoning.” 2022. https://leaderboard.allenai.org/arc/submissions/about.

Abebe, Rediet, Moritz Hardt, Angela Jin, John Miller, Ludwig Schmidt, and Rebecca Wexler. 2022. “Adversarial Scrutiny of Evidentiary Statistical Software.” In *2022 ACM Conference on Fairness, Accountability, and Transparency*, 1733–46. FAccT ’22. New York, NY, USA: Association for Computing Machinery. https://doi.org/10.1145/3531146.3533228.

Abid, Abubakar, Maheen Farooqi, and James Zou. 2021. “Persistent Anti-Muslim Bias in Large Language Models.” arXiv. https://doi.org/10.48550/arXiv.2101.05783.

Acemoglu, Daron. 2024. “Get Ready for the Great AI Disappointment.” *Wired*, 2024. https://www.wired.com/story/get-ready-for-the-great-ai-disappointment/.

Adler, Melissa. 2017. *Cruising the Library*. Fordham University Press. https://doi.org/10.2307/j.ctt1xhr79m.

Ager, Simon. 2023. “Omniglot - the Online Encyclopedia of Writing Systems and Languages.” 2023. https://omniglot.com/.

AI Advantage, Igor. 2024. “The AI Advantage.” YouTube. 2024. https://www.youtube.com/channel/UCHhYXsLBEVVnbvsq57n1MTQ.

AICommunity. 2024. “Community - AI Advantage.” 2024. https://myaiadvantage.com/community, https://myaiadvantage.com/community.

Akter, Syeda Nahida, Zichun Yu, Aashiq Muhamed, Tianyue Ou, Alex Bäuerle, Ángel Alexander Cabrera, Krish Dholakia, Chenyan Xiong, and Graham Neubig. 2023. “An In-Depth Look at Gemini’s Language Abilities.” arXiv. http://arxiv.org/abs/2312.11444.

Akyürek, Ekin, Dale Schuurmans, Jacob Andreas, Tengyu Ma, and Denny Zhou. 2022. “What Learning Algorithm Is In-Context Learning? Investigations with Linear Models.” arXiv. http://arxiv.org/abs/2211.15661.

Al Badi, Waleed, Laurie Alvandian, Anna Au, Magdalena Gomulka, Esther Bravo Govea, Louise-Anne Charles, Fatima Oury Sow Gueye, et al. 2023. “IFLA Trend Report 2022 Update.” https://repository.ifla.org/handle/123456789/2456.

Alammar, Jay. 2019. “The Illustrated Word2vec.” 2019. http://jalammar.github.io/illustrated-word2vec/.

Algorithmic Justice League. 2022. “Algorithmic Justice League - Unmasking AI Harms and Biases.” 2022. https://www.ajl.org/.

Alpert-Abrams, Hannah. 2016. “Machine Reading the Primeros Libros” 10 (4). http://www.digitalhumanities.org/dhq/vol/10/4/000268/000268.html.

Altman, Sam, dir. 2023. *OpenAI DevDay, Opening Keynote*. https://www.youtube.com/watch?v=U9mJuUkhUzk.

Amatriain, Xavier. 2023. “Transformer Models: An Introduction and Catalog.” arXiv.Org. 2023. https://arxiv.org/abs/2302.07730v2.

American Association of Law Libraries. 2019. “AALL Ethical Principles.” AALL. 2019. https://www.aallnet.org/advocacy/government-relations/recommended-guidelines/aall-ethical-principles/.

American Library Association. 2006. “Privacy: An Interpretation of the Library Bill of Rights.” https://www.ala.org/advocacy/intfreedom/librarybill/interpretations/privacy.
———. 2007. “Types of Libraries.” Text. Education & Careers. 2007. https://www.ala.org/educationcareers/careers/librarycareerssite/typesoflibraries.
———. 2008. “Office for Intellectual Freedom.” Text. About ALA. 2008. https://www.ala.org/aboutala/offices/oif.
———. 2018. “Facial Recognition.” Text. Tools, Publications & Resources. 2018. https://www.ala.org/tools/future/trends/facialrecognition.
———. 2021. “Professional Ethics and Code of Ethics.” Text. Tools, Publications & Resources. 2021. https://www.ala.org/tools/ethics.
Amodei, Dario, Danny Hernandez, Girish Sastry, Jack Clark, Greg Brockman, and Ilya Sutskever. 2019. “AI and Compute.” OpenAI. 2019. https://openai.com/blog/ai-and-compute/.
anc. 2023. “The Open American National Corpus.” 2023. https://anc.org/.
Angwin, Julia, and Jeff Larson. 2016. “Machine Bias.” Text/html. ProPublica. May 23, 2016. https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing.
ANSI/NISO, National Information Standards Organization. 2010. “Guidelines for the Construction, Format, and Management of Monolingual Controlled Vocabularies.” National Information Standards Organizaation. https://groups.niso.org/apps/group_public/download.php/12591/z39-19-2005r2010.pdf.
Anthropic. 2024. “Meet Claude.” 2024. https://www.anthropic.com/claude.
Araújo, Paula Carina de, Renata Cristina Gutierres Castanha, and Birger Hjørland. 2021. “Citation Indexing and Indexes.” *Knowledge Organization*, . Also available in ISKO Encyclopedia of Knowledge Organization, eds. Birger Hjørland and Claudio Gnoli, https://www.isko.org/cyclo/citation, 48 (1): 72–101.
ARCPrize. 2024. “ARC Prize - Official Guide.” ARC Prize. 2024. https://arcprize.org/guide.
Arlitsch, Kenning, and Bruce Newell. 2017. “Thriving in the Age of Accelerations: A Brief Look at the Societal Effects of Artificial Intelligence and the Opportunities for Libraries.” *Journal of Library Administration* 57 (7): 789–98. https://doi.org/10.1080/01930826.2017.1362912.
Aschenbrenner, Leopold. 2024. “Situational Awareness: The Decade Ahead.” 2024. https://situational-awareness.ai/.
Asemi, Asefeh, Andrea Ko, and Mohsen Nowkarizi. 2020. “Intelligent Libraries: A Review on Expert Systems, Artificial Intelligence, and Robot.” *Library Hi Tech* 39 (2): 412–34. https://doi.org/10.1108/LHT-02-2020-0038.
Ayre, Lori, and Jim Craner. 2018. “Algorithms: Avoiding the Implementation of Institutional Biases.” *Public Library Quarterly* 37 (3): 341–47. https://doi.org/10.1080/01616846.2018.1512811.
Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio. 2016. “Neural Machine Translation by Jointly Learning to Align and Translate.” arXiv. https://doi.org/10.48550/arXiv.1409.0473.

Bapna, Ankur, Isaac Caswell, Julia Kreutzer, Orhan Firat, Daan van Esch, Aditya Siddhant, Mengmeng Niu, et al. 2022. “Building Machine Translation Systems for the Next Thousand Languages.” arXiv. https://doi.org/10.48550/arXiv.2205.03983.
Barité, Mario. 2018. “Literary Warrant (IEKO).” 2018. https://www.isko.org/cyclo/literary_warrant.
Barocas, Solon, Kate Crawford, Aaron Shapiro, and Hanna Wallach. 2017. “The Problem with Bias: From Allocative to Representational Harms in Machine Learning. Special Interest Group for Computing, Information and Society (SIGCIS) (2017).”
Beer, David. 2017. “The Social Power of Algorithms.” *Information, Communication & Society* 20 (1): 1–13. https://doi.org/10.1080/1369118X.2016.1216147.
Bender, Emily M., Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. “On the Dangers of Stochastic Parrots: Can Language Models Be Too Big? 🦜.” In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, 610–23. FAccT ’21. New York, NY, USA: Association for Computing Machinery. https://doi.org/10.1145/3442188.3445922.
Benjamin, Garfield. 2022. “#FuckTheAlgorithm: Algorithmic Imaginaries and Political Resistance.” In *2022 ACM Conference on Fairness, Accountability, and Transparency*, 46–57. FAccT ’22. New York, NY, USA: Association for Computing Machinery. https://doi.org/10.1145/3531146.3533072.
Berman, Erin. 2018. “Big Brother Is Watching You: The Ethical Role of Libraries and Big Data.” *Choose Privacy Every Day* (blog). 2018. https://chooseprivacyeveryday.org/the-ethical-role-of-libraries-and-big-data/.
Berman, Matthew. 2024. “Intro to RAG for AI (Retrieval Augmented Generation).” 2024. https://www.youtube.com/watch?v=Y08Nn23o_mY.
Berman, Sanford. 1971. *Prejudices and Antipathies: A Tract on the LC Subject Heads Concerning People*. Jefferson, North Carolina: McFarland & Company, Inc.
———. 2000. “Review of Bowker, Geoffrey C. and Star, Susan Leigh. ‘Sorthing [Sic] Things out: Classification and Its Consequences.’” *Progressive Librarian* 17.
Berry, John N. 1998. “Choosing Sides: The Impasse between Prudes and Purists Has Forced the Issue.” *Library Journal* 123 (4): 6–7.
Bhatia, Aatish. 2023. “How Can an A.I. Learn to Write? Choose a Famous Author, and We’ll Show You.” *The New York Times*, 2023, sec. The Upshot. https://www.nytimes.com/interactive/2023/04/26/upshot/gpt-from-scratch.html.
Biosphere. 2022. “Home | Biosphere 2.” 2022. https://biosphere2.org/.
Blei, David M, Andrew Y. Ng, and Michael I. Jordan. 2003. “Latent Dirichlet Allocation.” *Journal of Machine Learning Research* 3:993–1022.
Blodgett, Su Lin, Solon Barocas, Hal Daumé III, and Hanna Wallach. 2020. “Language (Technology) Is Power: A Critical Survey of ‘Bias’ in NLP.” In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 5454–76. Online: Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.acl-main.485.
Bobrow, Daniel G, Ronald M Kaplan, Martin Kay, Donald A Norman, Henry Thompson, and Terry Winograd. 1977. “GUS, A Frame-Driven Dia|og System,” 19.

Bolukbasi, Tolga, Kai-Wei Chang, James Y Zou, Venkatesh Saligrama, and Adam T Kalai. 2016. “Man Is to Computer Programmer as Woman Is to Homemaker? Debiasing Word Embeddings.” In , 9. Barcelona, Spain.
Bommasani, Rishi, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, et al. 2022. “On the Opportunities and Risks of Foundation Models.” arXiv. https://doi.org/10.48550/arXiv.2108.07258.
Bommasani, Rishi, Kevin Klyman, Shayne Longpre, Sayash Kapoor, Nestor Maslej, Betty Xiong, Daniel Zhang, and Percy Liang. 2023. “The Foundation Model Transparency Index.” arXiv. https://doi.org/10.48550/arXiv.2310.12941.
Bourg, Chris. 2017. “What Happens to Libraries and Librarians When Machines Can Read All the Books?” *Feral Librarian* (blog). 2017. https://chrisbourg.wordpress.com/2017/03/16/what-happens-to-libraries-and-librarians-when-machines-can-read-all-the-books/.
Bowker, Geoffrey C., and Susan Leigh Star. 2000. *Sorting Things out: Classification and Its Consequences*. Cambridge, MA: The MIT Press.
Briggs, James, and Franciso Ingham. 2022. “LangChain AI Handbook.” Pinecone. 2022. https://www.pinecone.io/learn/langchain/.
British Library. 2020. “Digitised Manuscripts Harley MS 7368.” 2020. https://www.bl.uk/manuscripts/FullDisplay.aspx?ref=Harley_MS_7368.
Brooks, Frederick P. 1975. *The Mythical Man-Month*. 1st ed. Addison-Wesley.
Brown, Tom B., Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, et al. 2020. “Language Models Are Few-Shot Learners.” arXiv. https://doi.org/10.48550/arXiv.2005.14165.
Brunon-Ernst, Anne, ed. 2012. *Beyond Foucault: New Perspectives on Bentham’s Panopticon*. https://www.routledge.com/Beyond-Foucault-New-Perspectives-on-Benthams-Panopticon/Brunon-Ernst/p/book/9780754668435.
Budds, Diana, Diana Budds, and Diana Budds. 2017. “Biased AI Is A Threat To Civil Liberties. The ACLU Has A Plan To Fix It.” Fast Company. July 25, 2017. https://www.fastcompany.com/90134278/biased-ai-is-a-threat-to-civil-liberty-the-aclu-has-a-plan-to-fix-it.
Buolamwini, Joy. 2016. “The Coded Gaze.” AJL -ALGORITHMIC JUSTICE LEAGUE. November 6, 2016. https://medium.com/mit-media-lab/the-algorithmic-justice-league-3cc4131c5148.
———. 2019. “The Algorithmic Justice League.” *MIT MEDIA LAB* (blog). 2019. https://medium.com/mit-media-lab/the-algorithmic-justice-league-3cc4131c5148.
———. 2023. *Unmasking AI: My Mission to Protect What Is Human in the World of Machines*. https://www.penguinrandomhouse.com/books/670356/unmasking-ai-by-joy-buolamwini/.
Buolamwini, Joy, Vicente Ordóñez, Jamie Morgenstern, and Erik Learned-Miller. 2020. “Facial Recognition Technologies: A Primer.” https://assets.website-files.com/5e027ca188c99e3515b404b7/5ed1002058516c11edc66a14_FRTsPrimerMay2020.pdf.
Butcher, H.K., G.M. Bulechek, J.M. Docterman, and C.M. Wagner, eds. 2018. *Nursing Intervention Classification*.
c.ai. 2023. “Character.Ai.” Character.Ai. 2023. https://beta.character.ai/.

Caliskan, Aylin. 2021. “Detecting and Mitigating Bias in Natural Language Processing.” *Brookings* (blog). 2021. https://www.brookings.edu/research/detecting-and-mitigating-bias-in-natural-language-processing/.
Caliskan, Aylin, Joanna J. Bryson, and Arvind Narayanan. 2017. “Semantics Derived Automatically from Language Corpora Contain Human-like Biases.” *Science* 356 (6334): 183–86. https://doi.org/10.1126/science.aal4230.
Carlson, Jake, and Lisa Johnston. 2015. *Data Information Literacy: Librarians, Data, and the Education of a New Generation of Researchers*. http://public.eblib.com/choice/PublicFullRecord.aspx?p=2039088.
Cartter, Eileen. 2023. “The Pope Francis Puffer Photo Was Real in Our Hearts.” GQ. 2023. https://www.gq.com/story/pope-puffer-jacket-midjourney-ai-meme.
Casscells, W., A. Schoenberger, and T. B. Graboys. 1978. “Interpretation by Physicians of Clinical Laboratory Results.” *The New England Journal of Medicine* 299 (18): 999–1001. https://doi.org/10.1056/NEJM197811022991808.
Cauchy, Louis Augustin. 1847. “Méthode Générale Pour La Résolution Des Systémes d’équations Simultanées.” *Compte Rendu à l’Académie Des Sciences* 25:536–38.
CCP. 2020. “Center for Creative Photography.” Center for Creative Photography. 2020. https://ccp.arizona.edu/home.
Chan, Lois Mai. 2007. *Cataloging and Classification: An Introduction*. 3rd ed. Lanham, Maryland: The Scarecrow Press, Inc.
Chan, Lois Mai, Phyllis A. Richmond, and Elaine Svenonius. 1985. “Principles of Book Classification: E. Wyndham Hulme: Editor’s Introduction".” In *Theory of Subject Analysis. Edited by Lois Mai Chan, Phyllis A. Richmond and Elaine Svenonius. Littleton, Colorado, 48-49*.
Chase, Harrison. (2022) 2022. “LangChain.” Python. https://github.com/hwchase17/langchain.
Chen, Xiongwen, and Dale G. Brockway. 2017. “Height-Diameter Relationships in Longleaf Pine and Four Swamp Tree Species.” *Journal of Plant Studies* 6 (2): 94–101. https://www.srs.fs.usda.gov/pubs/54612.
Cherian, Anoop, Kuan-Chuan Peng, Suhas Lohit, Kevin A. Smith, and Joshua B. Tenenbaum. 2023. “Are Deep Neural Networks SMARTer than Second Graders?” arXiv. https://doi.org/10.48550/arXiv.2212.09993.
Chiang, Wei-Lin, Lianmin Zheng, Ying Sheng, Anastasios Nikolas Angelopoulos, Tianle Li, Dacheng Li, Hao Zhang, et al. 2024. “Chatbot Arena: An Open Platform for Evaluating LLMs by Human Preference.” arXiv. https://doi.org/10.48550/arXiv.2403.04132.
Cho, Kyunghyun, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. “Learning Phrase Representations Using RNN Encoder–Decoder for Statistical Machine Translation.” In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1724–34. Doha, Qatar: Association for Computational Linguistics. https://doi.org/10.3115/v1/D14-1179.
CHOICE Media Channel, dir. 2022. *Artificial Intelligence in Academic Libraries: How New AI Services Can Support Your Library Users*. https://www.youtube.com/watch?v=Ssg-sKLIq0k.
Chollet, François. 2019. “On the Measure of Intelligence.” arXiv. https://doi.org/10.48550/arXiv.1911.01547.

Chomsky, Noam, and Ramin Mirfakhraie. 2023. “ChatGPT and Human Intelligence: Noam Chomsky Responds to Critics | MR Online.” 2023. https://mronline.org/2023/04/24/chatgpt-and-human-intelligence-noam-chomsky-responds-to-critics/.
Chomsky, Noam, Ian Roberts, and Jeffrey Watumull. 2023. “Opinion | Noam Chomsky: The False Promise of ChatGPT.” *The New York Times*, 2023, sec. Opinion. https://www.nytimes.com/2023/03/08/opinion/noam-chomsky-chatgpt-ai.html.
Chowdhury, Ruman. 2023. “Rumman Chowdhury, Ph.D.” Rumman Chowdhury, Ph.D. 2023. http://www.rummanchowdhury.com.
Clariant Creative Agency. 2022. “Your Guide to Pillar Pages and Topic Clusters | Clariant Creative Agency.” 2022. https://www.clariantcreative.com/guide-to-pillar-pages-and-topic-clusters.
Clark, Jason A. (2018) 2022. “Algorithmic-Awareness.” Rich Text Format. https://github.com/jasonclark/algorithmic-awareness.
Coding Vidya. 2023. “Home | Coding Vidya - Best Computer Science Portal -.” 2023. https://codingvidya.com/.
Colby, Kenneth Mark, Franklin Dennis Hilf, Sylvia Weber, and Helena C Kraemer. 1972. “Turing-like Indistinguishability Tests for the Validation of a Computer Simulation of Paranoid Processes.” *Artificial Intelligence* 3:199–221. https://doi.org/10.1016/0004-3702(72)90049-5.
Colyer, Adrian. 2016. “The Amazing Power of Word Vectors | the Morning Paper.” 2016. https://blog.acolyer.org/2016/04/21/the-amazing-power-of-word-vectors/.
Common Crawl. 2022. “Common Crawl.” 2022. https://commoncrawl.org/.
Cook, John. 2023. “Cranky Uncle.” Cranky Uncle. 2023. https://crankyuncle.com/.
Corbett-Davies, Sam, and Sharad Goel. 2018. “The Measure and Mismeasure of Fairness: A Critical Review of Fair Machine Learning.” arXiv. https://doi.org/10.48550/arXiv.1808.00023.
Cordell, Ryan. 2020. “Machine Learning + Libraries.” LC Labs. Library of Congress. https://labs.loc.gov/static/labs/work/reports/Cordell-LOC-ML-report.pdf.
Corrado, Edward M. 2021. “Artificial Intelligence: The Possibilities for Metadata Creation.” *Technical Services Quarterly* 38 (4): 395–405. https://doi.org/10.1080/07317131.2021.1973797.
Costa, Ricky. 2023. “ChatGPT Cheat Sheet.” *Https://Neuralmagic.Com/*. https://www.kdnuggets.com/publications/sheets/ChatGPT_Cheatsheet_Costa.pdf.
Cox, Andrew M., and Suvodeep Mazumdar. 2022. “Defining Artificial Intelligence for Librarians.” *Journal of Librarianship and Information Science*, 09610006221142029. https://doi.org/10.1177/09610006221142029.
Cox, Andrew M., Stephen Pinfield, and Sophie Rutter. 2019. “The Intelligent Library: Thought Leaders’ Views on the Likely Impact of Artificial Intelligence on Academic Libraries.” *Library Hi Tech* 37 (3): 418–35. https://doi.org/10.1108/LHT-08-2018-0105.
Coyle, Karen. 2016. “FRBR Before and After.” 2016. http://kcoyle.net/beforeAndAfter/.
Crawford, Kate, dir. 2017. *The Trouble with Bias - NIPS 2017 Keynote - Kate Crawford #NIPS2017*. Neural Information Processing Systems. https://www.youtube.com/watch?v=fMym_BKWQzk.

———. 2022. “Lessons From The Panoptic Sort.” *International Journal of Communication*, no. 16, 1632–34.
Crichton, Gamal, Simon Baker, Yufan Guo, and Anna Korhonen. 2020. “Neural Networks for Open and Closed Literature-Based Discovery.” *PLOS ONE* 15 (5): e0232891. https://doi.org/10.1371/journal.pone.0232891.
Das, Kinnor, Clay J. Cockerell, Anant Patil, Paweł Pietkiewicz, Mario Giulini, Stephan Grabbe, and Mohamad Goldust. 2021. “Machine Learning and Its Application in Skin Cancer.” *International Journal of Environmental Research and Public Health* 18 (24): 13409. https://doi.org/10.3390/ijerph182413409.
Das, Rajesh Kumar, and Mohammad Sharif Ul Islam. 2021. “Application of Artificial Intelligence and Machine Learning in Libraries: A Systematic Review.” *arXiv:2112.04573 [Cs]*. http://arxiv.org/abs/2112.04573.
Data Information Literacy Project. 2023. “Data Information Literacy.” 2023. https://www.datainfolit.org/.
Davis, Wayne. 2019. “Implicature.” In *The Stanford Encyclopedia of Philosophy*, edited by Edward N. Zalta, Fall 2019. Metaphysics Research Lab, Stanford University. https://plato.stanford.edu/archives/fall2019/entries/implicature/.
Dean, Jeff. 2023. “Google Research, 2022 & beyond: Language, Vision and Generative Models.” 2023. https://ai.googleblog.com/2023/01/google-research-2022-beyond-language.html#GenerativeModels.
Deepankar, and Florian. 2023. “PromptPerfect - Elevate Your Prompts to Perfection with AI Prompt Engineering.” 2023. https://promptperfect.jina.ai/.
Dekker, Harrison, Angel Ferria, and Indrani Mandal. 2022. “URI Libraries’ AI Lab--Evolving to Meet the Needs of Students and Research Communities.”
Dempsey, Lorcan. 2023a. “Generative AI and Large Language Models: Background and Contexts.” LorcanDempsey.Net. 2023. https://www.lorcandempsey.net/intro-gen-ai/.
———. 2023b. “Generative AI, Scholarly and Cultural Language Models, and the Return of Content.” LorcanDempsey.Net. 2023. https://www.lorcandempsey.net/generative-ai-a-note-about-content/.
Desjardins, Jeff. 2017. “How Many Millions of Lines of Code Does It Take?” Visual Capitalist. February 8, 2017. https://www.visualcapitalist.com/millions-lines-of-code/.
Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. “BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding.” *arXiv:1810.04805 [Cs]*, May. http://arxiv.org/abs/1810.04805.
Dewland, Thomas A., Jeffrey E. Olgin, Eric Vittinghoff, and Gregory M. Marcus. 2013. “Incident Atrial Fibrillation Among Asians, Hispanics, Blacks, and Whites.” *Circulation* 128 (23): 2470–77. https://doi.org/10.1161/CIRCULATIONAHA.113.002449.
Dickson, Ben. 2023. “How to Customize LLMs like ChatGPT with Your Own Data and Documents - TechTalks.” 2023. https://bdtechtalks.com/2023/05/01/customize-chatgpt-llm-embeddings/.
Digital2030. 2022. “The Algorithm Literacy Project | Understanding Algorithms.” 2022. https://algorithmliteracy.org/.

Dizikes, Peter. 2011. “When the Butterfly Effect Took Flight.” MIT Technology Review. 2011. https://www.technologyreview.com/2011/02/22/196987/when-the-butterfly-effect-took-flight/.
Driess, Danny. 2023. “PaLM-E: An Embodied Multimodal Language Model.” 2023. https://ai.googleblog.com/2023/03/palm-e-embodied-multimodal-language.html.
Druga, Stefania, Sarah T. Vu, Eesh Likhith, and Tammy Qiu. 2019a. “Inclusive AI Literacy for Kids around the World.” In *Proceedings of FabLearn 2019*, 104–11. FL2019. New York, NY, USA: Association for Computing Machinery. https://doi.org/10.1145/3311890.3311904.
———. 2019b. “Inclusive AI Literacy for Kids around the World.” In *Proceedings of FabLearn 2019*, 104–11. FL2019. New York, NY, USA: Association for Computing Machinery. https://doi.org/10.1145/3311890.3311904.
Duhem, Pierre. 1914. *La Théorie Physique Son Objet et Sa Structure, 2nd Ed.,*. Translated by The Aim and Structure of Physical Theory English Translation Phillip Wiener. Paris: Chevalier et Rivière.
Economist. 2022. “Huge ‘Foundation Models’ Are Turbo-Charging AI Progress.” *The Economist*, 2022. https://www.economist.com/interactive/briefing/2022/06/11/huge-foundation-models-are-turbo-charging-ai-progress.
Econtalk. 2023. “Adam Mastroianni on Peer Review and the Academic Kitchen.” Econlib. 2023. https://www.econtalk.org/adam-mastroianni-on-peer-review-and-the-academic-kitchen/.
Egan, Margaret E., and Jesse H. Shera. 1952. “Foundations of a Theory of Bibliography.” https://www-jstor-org.ezproxy4.library.arizona.edu/stable/4304106#metadata_info_tab_contents.
Encyclopedia.com. 2019. “Encyclopedia.Com | Free Online Encyclopedia.” 2019. https://www.encyclopedia.com/.
ExLibris. 2019. “Artificial Intelligence in the Library: Advantages, Challenges and Tradition.” https://cdn2.hubspot.net/hubfs/2909474/Ex%20Libris%20Artificial%20Intelligence%20White%20Paper.pdf.
Fallis, Don. 2002. “Introduction: Social Epistemology and Information Science.” *Social Epistemology* 16 (1): 1–4. https://doi.org/10.1080/02691720210132752.
———. 2006. “Social Epistemology and Information Science.” In *Annual Review of Information Science and Technology*, edited by Blaise Cronin. Vol. 40.
Fedus, William, Barret Zoph, and Noam Shazeer. 2022. “Switch Transformers: Scaling to Trillion Parameter Models with Simple and Efficient Sparsity.” arXiv. https://doi.org/10.48550/arXiv.2101.03961.
Feigenbaum, E.A. 1989. “Toward the Library of the Future.” *Long Range Planning* 22 (1): 118–23. https://doi.org/10.1016/0024-6301(89)90059-9.
Fernandez, Peter. 2016. “‘Through the Looking Glass: Envisioning New Library Technologies’ How Artificial Intelligence Will Impact Libraries.” *Library Hi Tech News* 33 (5): 5–8. https://doi.org/10.1108/LHTN-05-2016-0024.
———. 2023. “‘Through the Looking Glass: Envisioning New Library Technologies’ AI-Text Generators as Explained by ChatGPT.” *Library Hi Tech News* 40 (3): 11–14. https://doi.org/10.1108/LHTN-02-2023-0017.

Firmani, D., Marco Maiorino, P. Merialdo, and Elena Nieddu. 2018. “Towards Knowledge Discovery from the Vatican Secret Archives. In Codice Ratio - Episode 1: Machine Transcription of the Manuscripts.” *KDD*. https://doi.org/10.1145/3219819.3219879.

Firmani, Donatella, Paolo Merialdo, and Marco Maiorino. 2017. “In Codice Ratio: Scalable Transcription of Vatican Registers.” 2017. https://ercim-news.ercim.eu/en111/special/in-codice-ratio-scalable-transcription-of-vatican-registers.

Fitch, Kent. 2023. “Searching for Meaning Rather Than Keywords and Returning Answers Rather Than Links.” *The Code4Lib Journal*, no. 57. https://journal.code4lib.org/articles/17443.

Fogg, B.J. 2003. *Persuasive Technology: Using Computers to Change What We Think and Do*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Frické, Martin. 1997. “Information Using Likeness Measures.” *Journal of the American Society for Information Science* 48:882–92.

———. 2012. *Logic and the Organization of Information*. New York: Springer.

———. 2015. “Big Data and Its Epistemology.” *Journal of the Association for Information Science and Technology* 66:651–61.

———. 2021. “Boolean Logic.” *Knowledge Organization* 48 (2): 177–91. https://doi.org/10.5771/0943-7444-2021-2-177.

Frické, Martin, Kay Mathiesen, and Don Fallis. 2000. “The Ethical Presuppositions behind the Library Bill of Rights.” *The Library Quarterly* 70 (4): 468–91. https://doi.org/10.1086/603218.

Friendly, Michael. 2007. “Gallery of Data Visualization.” 2007. https://www.datavis.ca/gallery/.

Froelich, Thomas. 2004. “A Brief History of Information Ethics.” BID 13 Desembre 2004. 2004.

Fu, Yao, Hao Peng, and Tushar Khot. 2023. “How Does GPT Obtain Its Ability? Tracing Emergent Abilities of Language Models to Their Sources.” 2023. https://yaofu.notion.site/How-does-GPT-Obtain-its-Ability-Tracing-Emergent-Abilities-of-Language-Models-to-their-Sources-b9a57ac0fcf74f30a1ab9e3e36fa1dc1.

Fuller, Steve. 1988. *Social Epistemology (Bloomington: Indiana University Press)*. Bloomington: Indiana University Press.

G2. 2023. “Best Bot Platforms Software.” G2. 2023. https://www.g2.com/categories/bot-platforms.

Gadd, Elizabeth. 2020. “AI-Based Citation Evaluation Tools: Good, Bad or Ugly?” *The Bibliomagician* (blog). 2020. https://thebibliomagician.wordpress.com/2020/07/23/ai-based-citation-evaluation-tools-good-bad-or-ugly/.

Gale, part of Cengage Group. 2023. “Library Marketing: Improve Outreach with Gale Analytics.” 2023. https://www.gale.com/databases/gale-analytics.

Gallagher, James, and Piotr Skalski. 2023. “First Impressions with GPT-4V(Ision).” Roboflow Blog. 2023. https://blog.roboflow.com/gpt-4-vision/.

Gandy Jr., Oscar H. 2021. *The Panoptic Sort: A Political Economy of Personal Information*. Second Edition, Second Edition. Oxford, New York: Oxford University Press.

Gao, Yunfan, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, and Haofen Wang. 2023. “Retrieval-Augmented Generation for Large Language Models: A Survey.” arXiv. https://doi.org/10.48550/arXiv.2312.10997.
Garnar, Martin, and Trina Magi, eds. 2021. *Intellectual Freedom Manual*. 10th ed. ALA Editions. https://www.alastore.ala.org/content/intellectual-freedom-manual-tenth-edition.
GDPR. 2018. “Art. 22 GDPR – Automated Individual Decision-Making, Including Profiling.” *General Data Protection Regulation (GDPR)* (blog). 2018. https://gdpr-info.eu/art-22-gdpr/.
Gee, Georgia. 2023. “Here Are the Stadiums That Are Keeping Track of Your Face.” *Slate*, 2023. https://slate.com/technology/2023/03/madison-square-garden-facial-recognition-stadiums-list.html.
Gesser, Avi, Robert Maddox, Anna Gressel, Mengyi Xu, Samuel J. Allaman, and Andres S. Gutierrez. 2022. “New Automated Decision-Making Laws: Four Tips for Compliance.” *Debevoise Data Blog* (blog). 2022. https://www.debevoisedatablog.com/2022/06/25/new-automated-decision-making-laws-four-tips-for-compliance/.
Gillies, Midge. 2020. “Amy Johnson – A Brief Biography – Amy Johnson Arts Trust.” 2020. http://amyjohnsonartstrust.co.uk/her-life/.
Github. 2022. “GitHub Copilot · Your AI Pair Programmer.” GitHub. 2022. https://github.com/features/copilot.
Glusac, Elaine. 2021. “Your Face Is, or Will Be, Your Boarding Pass.” *The New York Times*, 2021, sec. Travel. https://www.nytimes.com/2021/12/07/travel/biometrics-airports-security.html.
Goldman, Alvin I. 1999. *Knowledge in a Social World*. Oxford: Clarendon Press.
Goodman, Bryce, and Seth Flaxman. 2017. “European Union Regulations on Algorithmic Decision-Making and a ‘Right to Explanation.’” *AI Magazine* 38 (3): 50–57. https://doi.org/10.1609/aimag.v38i3.2741.
Google Cloud. 2023. “Speech-to-Text Supported Languages | Cloud Speech-to-Text Documentation | Google Cloud.” 2023. https://cloud.google.com/speech-to-text/docs/speech-to-text-supported-languages.
Google for Developers. 2022. “Machine Learning Crash Course.” 2022. https://developers.google.com/machine-learning/crash-course.
———. 2023. “Machine Learning Glossary: ML Fundamentals.” Google for Developers. 2023. https://developers.google.com/machine-learning/glossary/fundamentals.
Gozalo-Brizuela, Roberto, and Eduardo C. Garrido-Merchan. 2023. “ChatGPT Is Not All You Need. A State of the Art Review of Large Generative AI Models.” arXiv. https://doi.org/10.48550/arXiv.2301.04655.
Grant, Nico, and Kashmir Hill. 2023. “Google’s Photo App Still Can’t Find Gorillas. And Neither Can Apple’s.” *The New York Times*, 2023, sec. Technology. https://www.nytimes.com/2023/05/22/technology/ai-photo-labels-google-apple.html.
Griffey, Jason. 2019. “Artificial Intelligence and Machine Learning in Libraries.”
gwern. 2023. “Douglas Hofstadter Changes His Mind on Deep Learning & AI Risk.” https://www.lesswrong.com/posts/kAmgdEjq2eYQkB5PP/douglas-hofstadter-changes-his-mind-on-deep-learning-and-ai.

Hacking, Ian. 1999. *The Social Construction of What?* London: Harvard University Press.
HAI. 2023. “AI Index Report 2023 – Artificial Intelligence Index.” 2023. https://aiindex.stanford.edu/report/.
Hanson, Norwood Russell. 1958. *Patterns of Discovery*. Cambridge: Cambridge University Press.
Hardt, Moritz, Eric Price, and Nathan Srebro. 2016. “Equality of Opportunity in Supervised Learning [Preprint].” arXiv. https://doi.org/10.48550/arXiv.1610.02413.
Harper, Charlie. 2018. “Machine Learning and the Library or: How I Learned to Stop Worrying and Love My Robot Overlords.” *The Code4Lib Journal*, no. 41. https://journal.code4lib.org/articles/13671.
Harpring, Patricia. 2020. “Featuring the Getty Vocabularies,” 91.
Harris, Richard. 2002. “The deHavilland D.H.82 Tiger Moth and the Moth Family.” 2002. https://harris1.net/hold/av/avhist/a8/a8_moth.htm.
Hauptman, Robert. 1988. *Ethical Challenges in Librarianship*. Oryx.
———. 2002. *Ethics and Librarianship*. Jefferson, N.C.: McFarland.
Heatley, Louise M. 2023. “National Centre for Text Mining — NaCTEM.” XHTML. NaCTeM. 2023. https://www.nactem.ac.uk/index.php.
Heaven, Will Douglas. 2022. “Why Meta’s Latest Large Language Model Survived Only Three Days Online.” MIT Technology Review. 2022. https://www.technologyreview.com/2022/11/18/1063487/meta-large-language-model-ai-only-survived-three-days-gpt-3-science/.
Henderson, Leah. 2020. “The Problem of Induction.” In *The Stanford Encyclopedia of Philosophy*, edited by Edward N. Zalta, Spring 2020. Metaphysics Research Lab, Stanford University. https://plato.stanford.edu/archives/spr2020/entries/induction-problem/.
Hernan, Miguel. 2022. “Causal Diagrams: Draw Your Assumptions Before Your Conclusions.” edX. 2022. https://www.edx.org/course/causal-diagrams-draw-your-assumptions-before-your.
Hjørland, Birger. 2011. “Evidence-Based Practice: An Analysis Based on the Philosophy of Science.” *Journal of the American Society for Information Science and Technology* 62 (7): 1301–10. https://doi.org/10.1002/asi.21523.
Hofstadter, Douglas. 2018. “The Shallowness of Google Translate.” The Atlantic. 2018. https://www.theatlantic.com/technology/archive/2018/01/the-shallowness-of-google-translate/551570/.
Hollerith, Herman. 1889. Art of compiling statistics. United States US395782A, filed September 23, 1884, and issued 1889. https://patents.google.com/patent/US395782A/en.
Hond, Anne A. H. de, Marieke M. van Buchem, and Tina Hernandez-Boussard. 2022. “Picture a Data Scientist: A Call to Action for Increasing Diversity, Equity, and Inclusion in the Age of AI.” *Journal of the American Medical Informatics Association: JAMIA*, 2178–81. https://doi.org/10.1093/jamia/ocac156.
Hornik, Kurt, Maxwell Stinchcombe, and Halbert White. 1989. “Multilayer Feedforward Networks Are Universal Approximators.” *Neural Networks* 2 (5): 359–66. https://doi.org/10.1016/0893-6080(89)90020-8.

Howard, Jennifer. 2017. “What Happened to Google’s Effort to Scan Millions of University Library Books? - EdSurge News.” EdSurge. August 10, 2017. https://www.edsurge.com/news/2017-08-10-what-happened-to-google-s-effort-to-scan-millions-of-university-library-books.
Howson, Colin, and Peter Urbach. 2006. *Scientific Reasoning : The Bayesian Approach*. 3rd ed. Chicago: Open Court.
Huang, Austin, Suraj Subramanian, Jonathan Sum, Khalid Almubarak, and Stella Biderman. 2018. “The Annotated Transformer.” 2018. http://nlp.seas.harvard.edu/annotated-transformer/.
Hugging Face. 2023. “Hugging Face – The AI Community Building the Future.” 2023. https://huggingface.co/.
Huntington-Klein, Nick. 2022. *The Effect: An Introduction to Research Design and Causality | The Effect*. https://theeffectbook.net/index.html.
Huyen, Chip. 2023. “Building LLM Applications for Production.” 2023. https://huyenchip.com/2023/04/11/llm-engineering.html.
IFLA. 2012. “IFLA Code of Ethics for Librarians and Other Information Workers (Full Version) – IFLA.” 2012. https://www.ifla.org/publications/ifla-code-of-ethics-for-librarians-and-other-information-workers-full-version/.
———. 2020. “IFLA Statement on Libraries and Artificial Intelligence.” https://repository.ifla.org/handle/123456789/1646.
Imbler, Sabrina. 2021. “This Moth’s Name Is a Slur. Scientists Won’t Use It Anymore.” *The New York Times*, 2021, sec. Science. https://www.nytimes.com/2021/07/09/science/gypsy-moth-romani-entomological-society.html.
Iris.ai. 2023. “The Workspace Tools.” Iris.Ai - Your Researcher Workspace. 2023. https://iris.ai/features/.
Izquierdo, H. Andrés. 2022. “20 Artificial Intelligence and Text and Data Mining: Future Rules for Libraries?” In *Navigating Copyright for Libraries*, edited by Jessica Coates, Victoria Owen, and Susan Reilly, 497–540. De Gruyter Saur. https://doi.org/10.1515/9783110732009-022.
Jakeway, Eileen, Lauren Algee, Laurie Allen, Meghan Ferriter, Jaime Mears, Abigail Potter, and Kate Zwaard. 2020. “Machine Learning + Libraries Summit Event Summary.” LC Labs Digital Strategy Directorate.
Jo, Eun Seo, and Timnit Gebru. 2020. “Lessons from Archives: Strategies for Collecting Sociocultural Data in Machine Learning [Conference].” In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, 306–16. FAT* ’20. New York, NY, USA: Association for Computing Machinery. https://doi.org/10.1145/3351095.3372829.
Johansen, Johanna, Tore Pedersen, and Christian Johansen. 2021. “Studying Human-to-Computer Bias Transference.” *AI & SOCIETY*. https://doi.org/10.1007/s00146-021-01328-4.
Johnston, Lisa R., and Jon Jeffryes. 2015. “Teaching Civil Engineering Data Information Literacy Skills: An E-Learning Approach.” In *Data Information Literacy*, edited by Lisa R. Johnston and Jake Carlson, 149–78. Librarians, Data, and the Education of a New Generation of Researchers. Purdue University Press. https://www.jstor.org/stable/j.ctt6wq2vh.12.

Jurafsky, Dan, and James H. Martin. 2023. “Speech and Language Processing.” 2023. https://web.stanford.edu/~jurafsky/slp3/.
Jurafsky, Daniel, and James H. Martin. 2021. “Chapter 24 Chatbots & Dialogue Systems.” In *Speech and Language Processing. Daniel Jurafsky & James H. Martin. Copyright © 2021. All Rights Reserved. Draft of December 29, 2021.* https://web.stanford.edu/~jurafsky/slp3/24.pdf.
Kahneman, Daniel. 2011. *Thinking, Fast and Slow*. Penguin Books.
Kamradt, Greg. 2023. “Data Independent - YouTube.” 2023. https://www.youtube.com/channel/UCyR2Ct3pDOeZSRyZH5hPO-Q.
———. (2023) 2023. “Learn LangChain.” Jupyter Notebook. https://github.com/gkamradt/langchain-tutorials.
Kaplan, Jared, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. “Scaling Laws for Neural Language Models.” arXiv. https://doi.org/10.48550/arXiv.2001.08361.
Karpathy, Andrej. 2023a. “[D] A Baby GPT.” Reddit Post. *R/MachineLearning*. www.reddit.com/r/MachineLearning/comments/12h1zld/d_a_baby_gpt/.
———. 2023b. “Deep Neural Nets: 33 Years Ago and 33 Years from Now.” 2023. https://karpathy.github.io/2022/03/14/lecun1989/.
———. (2022) 2023. “nanoGPT.” Python. https://github.com/karpathy/nanoGPT.
———. 2023c. “State of GPT.” Microsoft Build. 2023. https://build.microsoft.com/en-US/sessions/db3f4859-cd30-4445-a0cd-553c3304f8e2.
Katell, Michael, Meg Young, Bernease Herman, Dharma Dailey, Aaron Tam, Vivian Guetler, Corinne Binz, Daniella Raz, and P. M. Krafft. 2019. “An Algorithmic Equity Toolkit for Technology Audits by Community Advocates and Activists.” arXiv. https://doi.org/10.48550/arXiv.1912.02943.
Kayid, Amr, and Nils Reimers. 2022. “Bonjour. مرحبا. Guten Tag. Hola. Cohere’s Multilingual Text Understanding Model Is Now Available.” Context by Cohere. 2022. https://txt.cohere.com/multilingual/.
Kim, Bohyun. 2017. “AI-Powered Robots for Libraries: Exploratory Questions.” In . Wildau, Germany. https://library.ifla.org/id/eprint/2700/.
Kirwan Institute. 2017. “2017 State of the Science: Implicit Bias Review | Kirwan Institute for the Study of Race and Ethnicity.” 2017. https://kirwaninstitute.osu.edu/article/2017-state-science-implicit-bias-review.
Kitcher, Philip. 2002. “Veritistic Value and the Project of Social Epistemology.” Edited by Alvin I. Goldman. *Philosophy and Phenomenological Research* 64 (1): 191–98. https://www.jstor.org/stable/3071029.
Knapp, Jeff. 2021. “Library Guides: ‘Fake’ News: Resources for Fact-Checking.” PennState University Libraries. 2021. https://guides.libraries.psu.edu/c.php?g=620262&p=4319365.
Knowlton, Steven A. 2005. “Three Decades Since Prejudices and Antipathies: A Study of Changes in the Library of Congress Subject Headings.” *Cataloging & Classification Quarterly, Vol. 40(2) 2005* 40:123–29.
Knox, Emily J.M. 2023. *Foundations of Intellectual Freedom*. Chicago: ALA Neal-Schuman.
Kunder, Maurice de. 2022. “WorldWideWebSize.Com | The Size of the World Wide Web (The Internet).” 2022. https://www.worldwidewebsize.com/.

Kurzweil, Ray. 2005. *The Singularity Is Near: When Humans Transcend Biology*. 1st edition. New York: The Viking Press.
Kusner, Matt J., Joshua R. Loftus, Chris Russell, and Ricardo Silva. 2018. "Counterfactual Fairness." *arXiv:1703.06856 [Cs, Stat]*. http://arxiv.org/abs/1703.06856.
Lamont, Julian, and Christi Favor. 2017. "Distributive Justice." In *The Stanford Encyclopedia of Philosophy*, edited by Edward N. Zalta, Winter 2017. Metaphysics Research Lab, Stanford University. https://plato.stanford.edu/archives/win2017/entries/justice-distributive/.
Larkin, Jill H., and Herbert A. Simon. 1987. "Why a Diagram Is (Sometimes) Worth Ten Thousand Words." *Cognitive Science* 11:65–99.
LeCun, Yann, Corinna Cortes, and Chris Burges. 1998. "MNIST Handwritten Digit Database." 1998. http://yann.lecun.com/exdb/mnist/.
Lee, Benjamin Charles Germain, Jaime Mears, Eileen Jakeway, Meghan Ferriter, Chris Adams, Nathan Yarasavage, Deborah Thomas, Kate Zwaard, and Daniel S. Weld. 2020. "The Newspaper Navigator Dataset: Extracting And Analyzing Visual Content from 16 Million Historic Newspaper Pages in Chronicling America." https://doi.org/10.48550/arXiv.2005.01583.
Lemaréchal, Claude. 2012. "Cauchy and the Gradient Method." *Documenta Mathematica* Extra Volume ISMP:251–54.
Lewis, Patrick, Ludovic Denoyer, and Sebastian Riedel. 2019. "Unsupervised Question Answering by Cloze Translation." In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 4896–4910. Florence, Italy: Association for Computational Linguistics. https://doi.org/10.18653/v1/P19-1484.
Lewis, Patrick, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, et al. 2021. "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks." arXiv. https://doi.org/10.48550/arXiv.2005.11401.
Library Hi Tech News. 2023. "Special Issue on ChatGPT." *Library Hi Tech News*. 40 (3).
Library of Congress. 2022. "G.Pdf." 2022. https://www.loc.gov/aba/publications/FreeLCSH/G.pdf.
Lin, Chin-Yew. 2004. "ROUGE: A Package for Automatic Evaluation of Summaries." In *Text Summarization Branches Out*, 74–81. Barcelona, Spain: Association for Computational Linguistics. https://aclanthology.org/W04-1013.
Litsey, Ryan, and Weston Mauldin. 2018. "Knowing What the Patron Wants: Using Predictive Analytics to Transform Library Decision Making." *The Journal of Academic Librarianship* 44 (1): 140–44. https://doi.org/10.1016/j.acalib.2017.09.004.
Liu, Rob. 2019. "Review of Weapons of Math Destruction Summary." 2019. https://lifeclub.org/books/weapons-of-math-destruction-cathy-oneil-review-summary.
Loo, Joshua. 2024. "System 1 and System 2 Thinking." The Decision Lab. 2024. https://thedecisionlab.com/reference-guide/philosophy/system-1-and-system-2-thinking.

Lorang, Elizabeth, Leen-Kiat Soh, Yi Liu, and Chulwoo Pack. 2020. “Digital Libraries, Intelligent Data Analytics, and Augmented Description: A Demonstration Project.” *Faculty Publications, UNL Libraries*. https://digitalcommons.unl.edu/libraryscience/396.
Lun, H.P. 1958. “The Automatic Creation of Literature Abstracts,” 159–65.
Lund, Brady D., and Ting Wang. 2023. “Chatting about ChatGPT: How May AI and GPT Impact Academia and Libraries?” *Library Hi Tech News* 40 (3): 26–29. https://doi.org/10.1108/LHTN-01-2023-0009.
Luong, Thang, Eugene Brevdo, and Rui Zhao. (2017) 2019. “Neural Machine Translation (Seq2seq) Tutorial.” Python. tensorflow. https://github.com/tensorflow/nmt.
Manning, Christopher D., Prabhakar Raghavan, and Hinrich Schütze. 2009. “Introduction to Information Retrieval.” 2009. https://nlp.stanford.edu/IR-book/information-retrieval-book.html.
Mao, Yuqing, and Zhiyong Lu. 2017. “MeSH Now: Automatic MeSH Indexing at PubMed Scale via Learning to Rank.” *Journal of Biomedical Semantics* 8 (April):15. https://doi.org/10.1186/s13326-017-0123-3.
Markowitz, Dale. 2022. “Meet AI’s Multitool: Vector Embeddings.” Google Cloud Blog. 2022. https://cloud.google.com/blog/topics/developers-practitioners/meet-ais-multitool-vector-embeddings.
McCulloch, Warren, and Walter Pitts. 1943. “A Logical Calculus of the Ideas Immanent in Nervous Activity.” *Bulletin of Mathematical Biophysics* 5:115–33.
McKenzie, Lindsay. 2018. “A New Home for AI: The Library.” Inside Higher Ed. 2018. https://www.insidehighered.com/news/2018/01/17/rhode-island-hopes-putting-artificial-intelligence-lab-library-will-expand-ais-reach.
McNeal, Michele L., and David Newyear. 2013. “Chapter 1: Introducing Chatbots in Libraries.” *Library Technology Reports* 49 (8): 5–10. https://journals.ala.org/index.php/ltr/article/view/4504.
Meszaros, Evan, and Mandi Goodsett. 2022. “Debunking & Prebunking: Strategies for Librarians to Eradicate Misinformation.” *American Library Association Annual Conference*. https://engagedscholarship.csuohio.edu/msl_facpub/183.
Meta. 2023. “Preserving the World’s Language Diversity Through AI.” *Meta* (blog). 2023. https://about.fb.com/news/2023/05/ai-massively-multilingual-speech-technology/.
———. 2024. “Meta Llama.” Meta Llama. 2024. https://llama.meta.com/.
Metz, Rachel. 2024. “OpenAI Scale Ranks Progress Toward ‘Human-Level’ Problem Solving.” *Bloomberg.Com*, 2024. https://www.bloomberg.com/news/articles/2024-07-11/openai-sets-levels-to-track-progress-toward-superintelligent-ai.
Mikolov, Tomas, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. “Efficient Estimation of Word Representations in Vector Space.” arXiv. http://arxiv.org/abs/1301.3781.
Mikolov, Tomas, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. “Distributed Representations of Words and Phrases and Their Compositionality.” In *Advances in Neural Information Processing Systems*. Vol. 26. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2013/hash/9aa42b31882ec039965f3c4923ce901b-Abstract.html.

Mikolov, Tomas, Wen-tau Yih, and Geoffrey Zweig. 2013. “Linguistic Regularities in Continuous Space Word Representations.” In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 746–51. Atlanta, Georgia: Association for Computational Linguistics. https://aclanthology.org/N13-1090.
Mill, John Stuart. 1869. “II. Of the Liberty of Thought and Discussion.” In *On Liberty*, edited by John Stuart Mill. London: Longman, Roberts & Green.
Miller, Johnathan. 2020. “The New Library User: Machine Learning.”
Minsky, Marvin, and Seymour Papert. 1969. *Perceptrons: An Introduction to Computational Geometry*.
Mishra, Prakhar. 2021. “10 Popular Keyword Extraction Algorithms in Natural Language Processing.” *MLearning.Ai* (blog). 2021. https://medium.com/mlearning-ai/10-popular-keyword-extraction-algorithms-in-natural-language-processing-8975ada5750c.
Mittelsteadt, Matthew G. 2023. “Artificial Intelligence: An Introduction for Policymakers | Mercatus Center.” 2023. https://www.mercatus.org/research/research-papers/artificial-intelligence-intro-for-policymakers.
Mollick, Ethan. 2023. “Working with AI: Two Paths to Prompting.” 2023. https://www.oneusefulthing.org/p/working-with-ai-two-paths-to-prompting.
Monigatti, Leonie. 2023. “Getting Started with LangChain: A Beginner’s Guide to Building LLM-Powered Applications.” Medium. 2023. https://towardsdatascience.com/getting-started-with-langchain-a-beginners-guide-to-building-llm-powered-applications-95fc8898732c.
Moreau, Erwan. 2023. “Literature-Based Discovery: Addressing the Issue of the Subpar Evaluation Methodology.” *Bioinformatics* 39 (2). https://doi.org/10.1093/bioinformatics/btad090.
Myint, Leslie, dir. 2020. *Key Structures in Causal Graphs*. https://www.youtube.com/watch?v=UA0vyBnzi9U.
Narang, Sharan, and Aakanksha Chowdhery. 2022. “Pathways Language Model (PaLM): Scaling to 540 Billion Parameters for Breakthrough Performance.” 2022. https://ai.googleblog.com/2022/04/pathways-language-model-palm-scaling-to.html.
Nawar, Tamer. 2021. “Veritism Refuted? Understanding, Idealization, and the Facts.” *Synthese* 198 (5): 4295–4313. https://doi.org/10.1007/s11229-019-02342-2.
Nguyen, Linh Cuong. 2020. “The Impact of Humanoid Robots on Australian Public Libraries.” *Journal of the Australian Library and Information Association* 69 (2): 130–48. https://doi.org/10.1080/24750158.2020.1729515.
Nielsen, Michael A. 2015. “Neural Networks and Deep Learning.” http://neuralnetworksanddeeplearning.com.
NLP-progress. 2022. “Tracking Progress in Natural Language Processing.” NLP-Progress. 2022. http://nlpprogress.com/.
Noble, Safiya. 2018. *Algorithms of Oppression: How Search Engines Reinforce Racism*. 1 edition. New York: NYU Press.
Nolan, Beatrice. 2022. “Artists Say AI Image Generators Are Copying Their Style to Make Thousands of New Images — and It’s Completely out of Their Control.”

Business Insider. 2022. https://www.businessinsider.com/ai-image-generators-artists-copying-style-thousands-images-2022-10.
Nori, Harsha, Samuel Jenkins, Paul Koch, and Rich Caruana. (2019) 2023. "InterpretML." C++. InterpretML. https://github.com/interpretml/interpret.
Norman, Donald A. 1993. *Things That Make Us Smart: Defending Human Attributes in the Age of the Machine*. Reading, MA: Addison-Wesley.
NuminaGroup. 2023. "Warehousing Encyclopedia." NuminaGroup. 2023. https://numinagroup.com/lp/warehousing-encyclopedia/.
Office of Educational Technology. 2023. "Artificial Intelligence and the Future of Teaching and Learning." Office of Educational Technology. 2023. https://tech.ed.gov/ai-future-of-teaching-and-learning/.
Olson, Hope A. 2000. "Difference, Culture and Change: The Untapped Potential of LCSH." *Cataloging & Classification Quarterly* 29:53–71.
———. 2002. *The Power to Name: Locating the Limits of Subject Representation in Libraries*. Boston: Kluwer.
On Large Language Models for Understanding Human Language Christopher Manning. 2022. https://www.youtube.com/watch?v=YfXc4OBDmnM.
O'Neil, Cathy. 2016. *Weapons of Math Destruction: How Big Data Increases Inequality and Threatens Democracy*. 1 edition. New York: Crown.
———, dir. 2018. *The Truth About Algorithms | Cathy O'Neil*. https://www.youtube.com/watch?v=heQzqX35c9A.
OpenAI. 2017. "Proximal Policy Optimization." OpenAI. 2017. https://openai.com/blog/openai-baselines-ppo/.
———. 2022a. "ChatGPT: Optimizing Language Models for Dialogue." OpenAI. 2022. https://openai.com/blog/chatgpt/.
———. 2022b. "Introducing Whisper." OpenAI. 2022. https://openai.com/blog/whisper/.
———. 2022c. "OpenAI." OpenAI. 2022. https://openai.com/.
———. 2023a. "GPT-4 Technical Report." GPT-4 Technical Report. 2023. https://cdn.openai.com/papers/gpt-4.pdf.
———. 2023b. "GPT-4V(Ision) System Card." 2023. https://cdn.openai.com/papers/GPTV_System_Card.pdf.
———. 2023c. "Guide to Prompt Engineering." 2023. https://platform.openai.com/docs/guides/prompt-engineering.
———. 2023d. "Introducing GPTs." 2023. https://openai.com/blog/introducing-gpts.
———. (2022) 2023. "OpenAI Cookbook." Jupyter Notebook. OpenAI. https://github.com/openai/openai-cookbook.
———. 2024. "Hello GPT-4o." 2024. https://openai.com/index/hello-gpt-4o/.
OpenAI Platform. 2024. "OpenAI Developer Platform." 2024. https://platform.openai.com.
Oppy, Graham, and David Dowe. 2021. "The Turing Test." In *The Stanford Encyclopedia of Philosophy*, edited by Edward N. Zalta, Winter 2021. Metaphysics Research Lab, Stanford University. https://plato.stanford.edu/archives/win2021/entriesuring-test/.
Ouyang, Long, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, et al. 2022. "Training Language Models to Follow Instructions with Human Feedback." arXiv. http://arxiv.org/abs/2203.02155.

Padilla, Thomas. 2019. “Responsible Operations: Data Science, Machine Learning, and AI in Libraries.”
Padilla, Thomas, Laurie Allen, Hannah Frost, Sarah Potvin, Elizabeth Russey Roke, and Stewart Varner. 2019. “Always Already Computational: Collections as Data: Final Report.” https://doi.org/10.5281/zenodo.3152935.
Pagano, Tiago P., Rafael B. Loureiro, Fernanda V. N. Lisboa, Rodrigo M. Peixoto, Guilherme A. S. Guimarães, Gustavo O. R. Cruz, Maira M. Araujo, et al. 2023. “Bias and Unfairness in Machine Learning Models: A Systematic Review on Datasets, Tools, Fairness Metrics, and Identification and Mitigation Methods.” *Big Data and Cognitive Computing* 7 (1): 15. https://doi.org/10.3390/bdcc7010015.
Pahwa, Nitish. 2023. “Silicon Valley’s Favorite New Toy Has a Risky Tradeoff.” *Slate*, 2023. https://slate.com/technology/2023/08/chatgpt-ai-arms-race-sustainability.html.
Patel, Dwarkesh. 2024. “Francois Chollet, Mike Knoop - LLMs Won’t Lead to AGI - $1,000,000 Prize to Find True Solution.” 2024. https://www.dwarkeshpatel.com/p/francois-chollet.
Pearl, Judea. 1995. “Causal Diagrams for Empirical Research.” *Biometrika* 82 (4): 669–88. https://doi.org/10.1093/biomet/82.4.669.
———. 2009a. “Causal Inference in Statistics : An Overview.” *Statistics Surveys* 3:96–146.
———. 2009b. *Causality: Models, Reasoning, and Inference*. 2nd ed. http://bayes.cs.ucla.edu/BOOK-2K/.
Peng, Ciyuan, Feng Xia, Mehdi Naseriparsa, and Francesco Osborne. 2023. “Knowledge Graphs: Opportunities and Challenges.” *Artificial Intelligence Review* 56 (11): 13071–102. https://doi.org/10.1007/s10462-023-10465-9.
Peroni, Silvio, and David Shotton. 2012. “FaBiO and CiTO: Ontologies for Describing Bibliographic Resources and Citations.” *Journal of Web Semantics* 17 (December):33–43. https://doi.org/10.1016/j.websem.2012.08.001.
Peters, Jay. 2020. “IBM Will No Longer Offer, Develop, or Research Facial Recognition Technology.” The Verge. 2020. https://www.theverge.com/2020/6/8/21284683/ibm-no-longer-general-purpose-facial-recognition-analysis-software.
Petroni, Fabio, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. “Language Models as Knowledge Bases?” In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2463–73. Hong Kong, China: Association for Computational Linguistics. https://doi.org/10.18653/v1/D19-1250.
Piantadosi, Steven. 2023. “Modern Language Models Refute Chomsky’s Approach to Language.” LingBuzz. https://lingbuzz.net/lingbuzz/007180.
Pichai, Sundar, and Demis Hassabis. 2023. “Introducing Gemini: Our Largest and Most Capable AI Model.” Google. 2023. https://blog.google/technology/ai/google-gemini-ai/.
Pickering, Ruth, Matthew Ismail, Daniel W. Hook, Simon J. Porter, Catherine Nicole Coleman, Michael A. Keller, James W. Weis, et al. 2022. *Artificial Intelligence in*

Libraries and Publishing. Michigan Publishing Services. https://doi.org/10.3998/mpub.12669942.
Pierce, John Robinson. 1980. *An Introduction to Information Theory : Symbols, Signals & Noise*. 2nd, rev. ed. New York: Dover Publications.
Plato. 380AD. "Meno." 380AD. http://classics.mit.edu/Plato/meno.html.
Poincaré, Henri. 1905. *Hypotheses in Physics*. Science and Hypothesis. London: Walter Scott Publishing.
Popper, Karl R. 1963. *Conjectures and Refutations*. London: Routledge and Kegan Paul.
———. 1968. "Epistemology Without a Knowing Subject." In *Studies in Logic and the Foundations of Mathematics*, 52:333–73. https://doi.org/10.1016/S0049-237X(08)71204-7.
Priem, Jason. 2013. "Beyond the Paper." *Nature* 495 (7442): 437–40. https://doi.org/10.1038/495437a.
Pritchard, Duncan, John Turri, and J. Adam Carter. 2022. "The Value of Knowledge." In *The Stanford Encyclopedia of Philosophy*, edited by Edward N. Zalta and Uri Nodelman, Fall 2022. Metaphysics Research Lab, Stanford University. https://plato.stanford.edu/archives/fall2022/entries/knowledge-value/.
"Project Implicit." 2011. 2011. https://implicit.harvard.edu/implicit/index.jsp.
Pushkin, Alexander. 1881. "Eugene Onegin." 1881. https://www.gutenberg.org/files/23997/23997-h/23997-h.htm.
Rabiner, L.R. 1989. "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition." *Proceedings of the IEEE* 77 (2): 257–86. https://doi.org/10.1109/5.18626.
Race, Technology, and Algorithmic Bias. 2019. https://www.radcliffe.harvard.edu/video/race-technology-and-algorithmic-bias-vision-justice.
Rainie, Lee, and Janna Anderson. 2017. "Code-Dependent: Pros and Cons of the Algorithm Age." *Pew Research Center: Internet, Science & Tech* (blog). 2017. https://www.pewresearch.org/internet/2017/02/08/code-dependent-pros-and-cons-of-the-algorithm-age/.
READ-COOP. 2021. "About Us." READ-COOP. 2021. https://readcoop.eu/about/.
Redi, Miriam. 2018. "How We're Using Machine Learning to Visually Enrich Wikidata." Wikimedia Foundation. 2018. https://wikimediafoundation.org/news/2018/03/14/machine-learning-visually-enriching-wikidata/.
Rees, David C., Thomas N. Williams, and Mark T. Gladwin. 2010. "Sickle-Cell Disease." *The Lancet* 376 (9757): 2018–31. https://doi.org/10.1016/S0140-6736(10)61029-X.
Reimers, Nils, and Jay Alammar. 2023. "The Embedding Archives: Millions of Wikipedia Article Embeddings in Many Languages." Context by Cohere. 2023. https://txt.cohere.com/embedding-archives-wikipedia/.
Reitz, Joan M. 2014. "ODLIS P." Online Dictionary for Library and Information Science. 2014. https://odlis.abc-clio.com/odlis_p.html.
Reuters. 2016. "New Zealand Passport Robot Tells Applicant of Asian Descent to Open Eyes." *Reuters*, 2016, sec. Technology News. https://www.reuters.com/article/us-newzealand-passport-error-idUSKBN13W0RL.

Rhody Today. 2017. “URI to Launch Artificial Intelligence Lab.” 2017. https://www.uri.edu/news/2017/12/uri-to-launch-artificial-intelligence-lab/.
Ridley, Michael, and Danica Pawlick-Potts. 2021a. “Algorithmic Literacy and the Role for Libraries.” *Information Technology and Libraries* 40 (2). https://doi.org/10.6017/ital.v40i2.12963.
———. 2021b. “Algorithmic Literacy and the Role for Libraries.” *Information Technology and Libraries* 40 (2). https://doi.org/10.6017/ital.v40i2.12963.
Roberts, David Lindsay. 2019. *Republic of Numbers*. https://doi.org/10.1353/book.67892.
Rohrer, Julia M. 2018. “Thinking Clearly About Correlations and Causation: Graphical Causal Models for Observational Data.” *Advances in Methods and Practices in Psychological Science* 1 (1): 27–42. https://doi.org/10.1177/2515245917745629.
Rolan, Gregory, Glen Humphries, Lisa Jeffrey, Evanthia Samaras, Tatiana Antsoupova, and Katharine Stuart. 2019. “More Human than Human? Artificial Intelligence in the Archive.” *Archives and Manuscripts* 47 (2): 179–203. https://doi.org/10.1080/01576895.2018.1502088.
Romero, Alberto. 2021. “GPT-3 — A Complete Overview.” Medium. 2021. https://towardsdatascience.com/gpt-3-a-complete-overview-190232eb25fd.
Rosenblatt, Frank. 1957. “The Perceptron A Perceiving and Recognizing Automaton (Project Para).” https://blogs.umass.edu/brain-wars/files/2016/03/rosenblatt-1957.pdf.
———. 1958. “The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain.” *Psychological Review* 65 (6): 386–408. https://doi.org/10.1037/h0042519.
Rosenblatt, Lucas, and R. Teal Witter. 2022. “Counterfactual Fairness Is Basically Demographic Parity.” arXiv. https://doi.org/10.48550/arXiv.2208.03843.
Rosenfeld, Louis, Peter Morville, and Jorge Arango. 2015. *Information Architecture: For the Web and Beyond*. O’Reilly Media, Inc.
Rowley, Jennifer. 2000. *Organising Knowledge: An Introduction to Managing Access to Information*. 3rd ed. Burlington, VT: Gower.
Roy, Kaushik, Vedant Khandelwal, Harshul Surana, Valerie Vera, Amit Sheth, and Heather Heckman. 2023. “GEAR-Up: Generative AI and External Knowledge-Based Retrieval Upgrading Scholarly Article Searches for Systematic Reviews.” arXiv. http://arxiv.org/abs/2312.09948.
Rubenstein, Paul K., Chulayuth Asawaroengchai, Duc Dung Nguyen, Ankur Bapna, Zalán Borsos, Félix de Chaumont Quitry, Peter Chen, et al. 2023. “AudioPaLM: A Large Language Model That Can Speak and Listen.” arXiv. https://doi.org/10.48550/arXiv.2306.12925.
Rumelhart, David E., Geoffrey E. Hinton, and Ronald J. Williams. 1986. “Learning Representations by Back-Propagating Errors.” *Nature* 323 (6088): 533–36. https://doi.org/10.1038/323533a0.
Russell, Bertrand. 1912. *The Problems of Philosophy*. https://www.gutenberg.org/files/5827/5827-h/5827-h.htm.
Rutkowski, Greg. 2023. “Greg Rutkowki: Artist.” 2023. https://rutkowski.artstation.com/.
Sadeh, Tamar. 2015. “From Search to Discovery.” *Bibliothek Forschung Und Praxis* 39 (2): 212–24. https://doi.org/10.1515/bfp-2015-0028.

Samuel, Arthur L. 1959. “Eight-Move Opening Utilizing Generalization Learning. (See Appendix B, Game G-43.1 Some Studies in Machine Learning Using the Game of Checkers. IBM Journal, 210–229.” In .
Sanderson, Grant, and 3Blue1Brown, dirs. 2017a. *But What Is a Neural Network? | Chapter 1, Deep Learning*. https://www.youtube.com/watch?v=aircAruvnKk.
———, dirs. 2017b. *Gradient Descent, How Neural Networks Learn | Chapter 2, Deep Learning*. https://www.youtube.com/watch?v=IHZwWFHWa-w.
Sanji, Majideh, Hassan Behzadi, and Gisu Gomroki. 2022. “Chatbot: An Intelligent Tool for Libraries.” *Library Hi Tech News* ahead-of-print. https://doi.org/10.1108/LHTN-01-2021-0002.
Saravia, Elvis. 2023. “Prompt Engineering Guide – Nextra.” 2023. https://www.promptingguide.ai/.
Scheines, Richard. 1997. “An Introduction to Causal Inference.” In *Causality in Crisis?*, 185–200. University of Notre Dame.
Schlicht, Matt, and Ben Parr. 2023. “Chatbots Magazine: The #1 Place to Learn about Chatbots.” Chatbots Magazine. 2023. https://chatbotsmagazine.com/.
Sebastian, Yakub, Eu-Gene Siew, and Sylvester O. Orimaye. 2017. “Emerging Approaches in Literature-Based Discovery: Techniques and Performance Review.” *The Knowledge Engineering Review* 32:e12. https://doi.org/10.1017/S0269888917000042.
Seff, Ari, dir. 2023. *How ChatGPT Is Trained*. https://www.youtube.com/watch?v=VPRSBzXzavo.
Shannon, C E. 1948. “A Mathematical Theory of Communication.” *The Bell System Technical Journal* 27:379–423, 623–56.
Shannon, Claude Elwood, and Warren Weaver. 1949. *The Mathematical Theory of Communication*. Urbana: University of Illinois Press.
Singhal, Amit. 2012. “Introducing the Knowledge Graph: Things, Not Strings.” Google. 2012. https://blog.google/products/search/introducing-knowledge-graph-things-not/.
Smalheiser, Neil R. 2017. “Rediscovering Don Swanson: The Past, Present and Future of Literature-Based Discovery.” *Journal of Data and Information Science (Warsaw, Poland)* 2 (4): 43–64. https://doi.org/10.1515/jdis-2017-0019.
Smith, Linda C. 1981. “Citation Analysis.” *Library Trends* 30:83–106.
Smith, Martha Montague. 1997. “Information Ethics.” In *Annual Review of Information Science and Technology*, 32:339–66.
Snow, Karen. 2017. “Defining, Assessing, and Rethinking Quality Cataloging.” *Cataloging & Classification Quarterly* 55 (7–8): 438–55. https://doi.org/10.1080/01639374.2017.1350774.
Society of American Archivists. 2020. “SAA Core Values Statement and Code of Ethics | Society of American Archivists.” 2020. https://www2.archivists.org/statements/saa-core-values-statement-and-code-of-ethics.
Somers, James. 2017. “Torching the Modern-Day Library of Alexandria.” The Atlantic. 2017. https://www.theatlantic.com/technology/archive/2017/04/the-tragedy-of-google-books/523320/.
Spivak, Nova, and Nick Slavin. 2023. “Arch Mission Foundation.” Arch Mission. 2023. https://archmission.org/.

Stamp, Mark. 2017. “A Revealing Introduction to Hidden Markov Models.” In *Introduction to Machine Learning with Applications in Information Security*, by Mark Stamp, 1st ed., 7–35. Chapman and Hall/CRC. https://doi.org/10.1201/9781315213262-2.
Stanford HAI. 2023a. “AI Will Transform Teaching and Learning. Let’s Get It Right.” Stanford HAI. 2023. https://hai.stanford.edu/news/ai-will-transform-teaching-and-learning-lets-get-it-right.
———. 2023b. “Generative AI: Perspectives from Stanford HAI.” 2023. https://hai.stanford.edu/sites/default/files/2023-03/Generative_AI_HAI_Perspectives.pdf.
Starmer, Josh, dir. 2020. *Neural Networks Pt. 2: Backpropagation Main Ideas*. https://www.youtube.com/watch?v=IN2XmBhILt4.
Starmer, Josh, and StatQuest, dirs. 2019. *Gradient Descent, Step-by-Step*. https://www.youtube.com/watch?v=sDv4f4s2SB8.
Strevens, Michael. 2013. “Looking into the Black Box.” Opinionator. 2013. https://archive.nytimes.com/opinionator.blogs.nytimes.com/2013/11/24/looking-into-the-black-box/.
Suber, Peter. 1999. “Paternalism.” 1999. https://dash.harvard.edu/bitstream/handle/1/4725017/suber_paternal.htm.
Sueiras, Jorge. 2021. “Continuous Offline Handwriting Recognition Using Deep Learning Models.” arXiv. https://doi.org/10.48550/arXiv.2112.13328.
Sulmont, Elisabeth, Elizabeth Patitsas, and Jeremy R. Cooperstock. 2019. “Can You Teach Me To Machine Learn?” In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*, 948–54. SIGCSE ’19. New York, NY, USA: Association for Computing Machinery. https://doi.org/10.1145/3287324.3287392.
Suresh, Harini, and John Guttag. 2021. “A Framework for Understanding Sources of Harm throughout the Machine Learning Life Cycle.” In *Equity and Access in Algorithms, Mechanisms, and Optimization*, 1–9. -- NY USA: ACM. https://doi.org/10.1145/3465416.3483305.
Sutskever, Ilya, Oriol Vinyals, and Quoc V Le. 2014. “Sequence to Sequence Learning with Neural Networks.” In *Advances in Neural Information Processing Systems*. Vol. 27. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2014/hash/a14ac55a4f27472c5d894ec1c3c743d2-Abstract.html.
Sutton, Richard S., and Andrew G. Barto. 2018. *Reinforcement Learning: An Introduction*. 2nd ed. http://incompleteideas.net/book/the-book-2nd.html.
Svenonius, Elaine. 1969. Review of *Review of Two Kinds of Power: An Essay on Bibliographical Control*, by Patrick Wilson. *The Library Quarterly: Information, Community, Policy* 39 (1): 112–14. http://www.jstor.org/stable/4305960.
———. 2003. “Design of Controlled Vocabularies.” In *Encyclopedia of Library and Information Science*, 822–38. New York: Marcel Dekker.
Swanson, D. R. 1986. “Undiscovered Public Knowledge.” *Library Quarterly* 56:103–18.
Tait, Elizabeth, and Cameron M Pierson. 2022. “Artificial Intelligence and Robots in Libraries: Opportunities in LIS Curriculum for Preparing the Librarians of Tomorrow.” *Journal of the Australian Library and Information Association* 71 (3): 256–74. https://doi.org/10.1080/24750158.2022.2081111.

Tashea, Jason. 2017. “Courts Are Using AI to Sentence Criminals. That Must Stop Now.” *Wired*, 2017. https://www.wired.com/2017/04/courts-using-ai-sentence-criminals-must-stop-now/.
Tay, Aaron. 2022. “List of Innovative Literature Mapping Tools | Aaron Tay’s Musings about Librarianship.” 2022. https://musingsaboutlibrarianship.blogspot.com/p/list-of-innovative-literature-mapping.html?view=classic.
Taylor, Arlene G. 2004. *The Organization of Information*. 2nd ed. Westport, Conn: Libraries Unlimited.
Taylor, Ross, Marcin Kardas, Guillem Cucurull, Thomas Scialom, Anthony Hartshorn, Elvis Saravia, Andrew Poulton, Viktor Kerkez, and Robert Stojnic. 2022. “Galactica: A Large Language Model for Science.” arXiv. https://doi.org/10.48550/arXiv.2211.09085.
Taylor, Wilson L. 1953. “‘Cloze Procedure’: A New Tool for Measuring Readability.” *Journalism Quarterly* 30 (4): 415–33. https://doi.org/10.1177/107769905303000401.
Teixeira, Lawrence. 2023. “The New Open AI GPT-4 Vision on ChatGPT: Bridging the Gap Between Text and Image Understanding.” 2023. https://medium.com/@lawrenceteixeira/the-new-open-ai-gpt-4-vision-on-chatgpt-bridging-the-gap-between-text-and-image-understanding-9337ed4c1a61.
Tella, Adeyinka. 2020. “Robots Are Coming to the Libraries: Are Librarians Ready to Accommodate Them?” *Library Hi Tech News* 37 (8): 13–17. https://doi.org/10.1108/LHTN-05-2020-0047.
Tella, Adeyinka, and Yusuf Ayodeji Ajani. 2022. “Robots and Public Libraries.” *Library Hi Tech News* 39 (7): 15–18. https://doi.org/10.1108/LHTN-05-2022-0072.
Tenney, Ian, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R. Thomas McCoy, Najoung Kim, et al. 2022. “What Do You Learn from Context? Probing for Sentence Structure in Contextualized Word Representations.” In . https://openreview.net/forum?id=SJzSgnRcKX.
The AI Advantage, dir. 2023. *100+ Insane ChatGPT Vision Use Cases*. https://www.youtube.com/watch?v=ywNNRzc7-T0.
Thilakaratne, Menasha, Katrina Falkner, and Thushari Atapattu. 2020. “A Systematic Review on Literature-Based Discovery: General Overview, Methodology, & Statistical Analysis.” *ACM Computing Surveys* 52 (6): 1–34. https://doi.org/10.1145/3365756.
Thoppilan, Romal, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, et al. 2022. “LaMDA: Language Models for Dialog Applications.” arXiv. https://doi.org/10.48550/arXiv.2201.08239.
Thunström, Almira Osmanovic. 2022. “We Asked GPT-3 to Write an Academic Paper about Itself—Then We Tried to Get It Published.” Scientific American. 2022. https://www.scientificamerican.com/article/we-asked-gpt-3-to-write-an-academic-paper-about-itself-then-we-tried-to-get-it-published/.
Turner, Ash. 2018. “How Many People Have Smartphones Worldwide (Oct 2022).” 2018. https://www.bankmycell.com/blog/how-many-phones-are-in-the-world.
Tversky, A. 1974. “Judgments under Uncertainty: Heuristics and Biases.” *Science* 185:1124-1131.

Tversky, Amos, and Daniel Kahneman. 1982. “Evidential Impact of Base Rates.” In *Judgement under Uncertainty: Heuristics and Biases*, edited by Daniel Kahneman, A Slovic, and Amos Tversky. Cambridge University Press.
UCL. 2018. “Transcribe Bentham.” Bentham Project. 2018. https://www.ucl.ac.uk/bentham-project/transcribe-bentham.
UCSF Office of Diversity and Outreach UCSF. 2022. “Unconscious Bias Training.” University of California: Office of Diversity and Outreach. 2022. https://diversity.ucsf.edu/programs-resources/training/unconscious-bias-training.
ujet.cx. 2022a. “Critical State of Automation in Customer Experience.” UJET. 2022. https://ujet.cx/resources/reports/critical-state-of-automation-customer-experience-2022-report-lp.
———. 2022b. “UJET Research Reveals Chatbots Increase Frustration for 80% of Consumers.” UJET. 2022. https://ujet.cx/press-releases/ujet-research-reveals-chatbots-increase-frustration.
University of Alberta Library. 2023. “Evidence Based Library and Information Practice.” 2023. https://journals.library.ualberta.ca/eblip/index.php/EBLIP.
Uszkoreit, Jakob. 2017. “Transformer: A Novel Neural Network Architecture for Language Understanding.” 2017. https://ai.googleblog.com/2017/08/transformer-novel-neural-network.html.
VanderWeele, Tyler J., and Nancy Staudt. 2011. “Causal Diagrams for Empirical Legal Research: A Methodology for Identifying Causation, Avoiding Bias and Interpreting Results.” *Law, Probability & Risk : A Journal of Reasoning under Uncertainty* 10 (4): 329–54. https://doi.org/10.1093/lpr/mgr019.
Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. “Attention Is All You Need.” arXiv. https://doi.org/10.48550/arXiv.1706.03762.
Vincze, Joseph. 2017. “Virtual Reference Librarians (Chatbots).” *Library Hi Tech News* 34 (4): 5–8. https://doi.org/10.1108/LHTN-03-2017-0016.
Von Hilgers, Philipp, and Amy M. Langville. 2006. “The Five Greatest Applications of Markov Chains.” 2006. http://langvillea.people.cofc.edu/MCapps7.pdf.
W3C Working Group. 2014. “RDF 1.1 Primer.” 2014. https://www.w3.org/TR/rdf11-primer/.
Wang, Angelina, Solon Barocas, Kristen Laird, and Hanna Wallach. 2022. “Measuring Representational Harms in Image Captioning.” In *2022 ACM Conference on Fairness, Accountability, and Transparency*, 324–35. FAccT ’22. New York, NY, USA: Association for Computing Machinery. https://doi.org/10.1145/3531146.3533099.
Wattenberg, Martin, Fernanda Viégas, and Moritz Hardt. 2022. “Attack Discrimination with Smarter Machine Learning.” 2022. https://research.google.com/bigpicture/attacking-discrimination-in-ml/.
Wei, Jason, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. “Chain-of-Thought Prompting Elicits Reasoning in Large Language Models.” arXiv. https://doi.org/10.48550/arXiv.2201.11903.

Wei, and Denny Zhou. 2022. “Language Models Perform Reasoning via Chain of Thought.” 2022. https://ai.googleblog.com/2022/05/language-models-perform-reasoning-via.html.
Weidinger, Laura, Jonathan Uesato, Maribeth Rauh, Conor Griffin, Po-Sen Huang, John Mellor, Amelia Glaese, et al. 2022. “Taxonomy of Risks Posed by Language Models.” In *2022 ACM Conference on Fairness, Accountability, and Transparency*, 214–29. FAccT ’22. New York, NY, USA: Association for Computing Machinery. https://doi.org/10.1145/3531146.3533088.
Weigert, Verena. 2020. “Chatbots in Libraries – Library Services.” 2020. https://libraryservices.jiscinvolve.org/wp/2020/09/chatbots-in-libraries/.
Weizenbaum, Joseph. 1966. “ELIZA—a Computer Program for the Study of Natural Language Communication between Man and Machine.” *Communications of the ACM* 9 (1): 36–45. https://doi.org/10.1145/365153.365168.
Wells, David. 2021. “Online Public Access Catalogues and Library Discovery Systems.” Text. 2021. https://www.isko.org/cyclo/opac#6.
Wenar, Leif. 2021. “Rights.” In *The Stanford Encyclopedia of Philosophy*, edited by Edward N. Zalta, Spring 2021. Metaphysics Research Lab, Stanford University. https://plato.stanford.edu/archives/spr2021/entries/rights/.
Wheatley, Amanda, and Sandy Hervieux. 2019. “Artificial Intelligence in Academic Libraries: An Environmental Scan.” *Information Services & Use* 39 (4): 347–56. https://doi.org/10.3233/ISU-190065.
Widyassari, Adhika Pramita, Supriadi Rustad, Guruh Fajar Shidik, Edi Noersasongko, Abdul Syukur, Affandy Affandy, and De Rosal Ignatius Moses Setiadi. 2020. “Review of Automatic Text Summarization Techniques & Methods.” *Journal of King Saud University - Computer and Information Sciences*. https://doi.org/10.1016/j.jksuci.2020.05.006.
Wikipedia. 2022a. “Algorithm.” In *Wikipedia*. https://en.wikipedia.org/w/index.php?title=Algorithm.
———. 2022b. “Algorithmic Bias.” In *Wikipedia*. https://en.wikipedia.org/w/index.php?title=Algorithmic_bias.
———. 2022c. “Artificial Linguistic Internet Computer Entity.” In *Wikipedia*. https://en.wikipedia.org/w/index.php?title=Artificial_Linguistic_Internet_Computer_Entity&oldid=1112755868.
———. 2022d. “False Positives and False Negatives.” In *Wikipedia*. https://en.wikipedia.org/w/index.php?title=False_positives_and_false_negatives&oldid=1088158900.
———. 2022e. “LaMDA.” In *Wikipedia*. https://en.wikipedia.org/w/index.php?title=LaMDA&oldid=1103832671.
———. 2022f. “Vatican Apostolic Archive.” In *Wikipedia*. https://en.wikipedia.org/w/index.php?title=Vatican_Apostolic_Archive&oldid=1095541005.
———. 2023a. “CAPTCHA.” In *Wikipedia*. https://en.wikipedia.org/w/index.php?title=CAPTCHA.
———. 2023b. “Confirmation Bias.” In *Wikipedia*. https://en.wikipedia.org/w/index.php?title=Confirmation_bias&oldid=1186068624.

———. 2023c. “Crowdsourcing.” In *Wikipedia*. https://en.wikipedia.org/w/index.php?title=Crowdsourcing&oldid=1131878886.
———. 2023d. “Evidence-Based Practice.” In *Wikipedia*. https://en.wikipedia.org/w/index.php?title=Evidence-based_practice&oldid=1141797027.
———. 2023e. “Explainable Artificial Intelligence.” In *Wikipedia*. https://en.wikipedia.org/w/index.php?title=Explainable_artificial_intelligence&oldid=1144112716.
———. 2023f. “Google Hummingbird.” In *Wikipedia*. https://en.wikipedia.org/w/index.php?title=Google_Hummingbird&oldid=1152805833.
———. 2023g. “LangChain.” In *Wikipedia*. https://en.wikipedia.org/w/index.php?title=LangChain.
———. 2023h. “Liberty Leading the People.” In *Wikipedia*. https://en.wikipedia.org/w/index.php?title=Liberty_Leading_the_People&oldid=1158901762.
———. 2023i. “Library.” In *Wikipedia*. https://en.wikipedia.org/w/index.php?title=Library&oldid=1140601084.
———. 2023j. “Literature-Based Discovery.” In *Wikipedia*. https://en.wikipedia.org/w/index.php?title=Literature-based_discovery&oldid=1140927264.
———. 2023k. “Open Archives Initiative Protocol for Metadata Harvesting.” In *Wikipedia*. https://en.wikipedia.org/w/index.php?title=Open_Archives_Initiative_Protocol_for_Metadata_Harvesting&oldid=1133365780.
———. 2023l. “PageRank.” In *Wikipedia*. https://en.wikipedia.org/w/index.php?title=PageRank&oldid=1165448388.
———. 2023m. “RankBrain.” In *Wikipedia*. https://en.wikipedia.org/w/index.php?title=RankBrain&oldid=1140034006.
———. 2023n. “Tay (Chatbot).” In *Wikipedia*. https://en.wikipedia.org/w/index.php?title=Tay_(chatbot).
———. 2023o. “The Library of Babel.” In *Wikipedia*. https://en.wikipedia.org/w/index.php?title=The_Library_of_Babel&oldid=1141372445.
———. 2023p. “Wikipedia:Artificial Intelligence.” In *Wikipedia*. https://en.wikipedia.org/w/index.php?title=Wikipedia:Artificial_intelligence&oldid=1157714616.
———. 2024. “Her.” In *Wikipedia*. https://en.wikipedia.org/w/index.php?title=Her_(film)&oldid=1237911810.
Wilburn, Brad. 1999. “Spice for the Good Life.” *Issues in Ethics* 10.
Wilson, P. 1968. *Two Kinds of Power: An Essay on Bibliographical Control*. Berkeley: University of California Press.
Wolfe, Matt. 2023. “Future Tools - Find The Exact AI Tool For Your Needs.” 2023. https://www.futuretools.io/.
Wolford, Ben. 2018. “What Is GDPR, the EU’s New Data Protection Law?” GDPR.Eu. 2018. https://gdpr.eu/what-is-gdpr/.

Wolfram. 2023a. “Wolfram Mathematica: Modern Technical Computing.” 2023. https://www.wolfram.com/mathematica/.
Wolfram, Stephen. 2023b. “ChatGPT Gets Its ‘Wolfram Superpowers’!” 2023. https://writings.stephenwolfram.com/2023/03/chatgpt-gets-its-wolfram-superpowers/.
Wolkoff, K. N. 1996. “The Problem of Holocaust Denial Literature in Libraries.” *Library Trends* 45:87–96.
Wood, Barbara, and David Evans. 2018. “Librarians’ Perceptions of Artificial Intelligence and Its Potential Impact on the Profession.” *Computers in Libraries* 38 (1). https://www.researchgate.net/publication/322977069_Librarians'_Perceptions_of_Artificial_Intelligence_and_Its_Potential_Impact_on_the_Profession.
Writer, Beta. 2019. *Lithium-Ion Batteries*. https://link.springer.com/book/10.1007/978-3-030-16800-1.
Xiang, Chloe. 2023. “OpenAI’s GPT-4 Is Closed Source and Shrouded in Secrecy.” Vice. 2023. https://www.vice.com/en/article/ak3w5a/openais-gpt-4-is-closed-source-and-shrouded-in-secrecy.
Xu, Binfeng, Zhiyuan Peng, Bowen Lei, Subhabrata Mukherjee, Yuchen Liu, and Dongkuan Xu. 2023. “ReWOO: Decoupling Reasoning from Observations for Efficient Augmented Language Models.” arXiv. https://doi.org/10.48550/arXiv.2305.18323.
Yang, Zhengyuan, Linjie Li, Kevin Lin, Jianfeng Wang, Chung-Ching Lin, Zicheng Liu, and Lijuan Wang. 2023. “The Dawn of LMMs: Preliminary Explorations with GPT-4V(Ision).” arXiv. https://doi.org/10.48550/arXiv.2309.17421.
Zellers, Rowan, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. “HellaSwag: Can a Machine Really Finish Your Sentence?” arXiv. http://arxiv.org/abs/1905.07830.
Zeng, Marcia Lei. 2005. “Construction of Controlled Vocabularies, A Primer (Based on Z39.19).” 2005.
Zewe, Adam. 2023. “Solving a Machine-Learning Mystery.” MIT News | Massachusetts Institute of Technology. 2023. https://news.mit.edu/2023/large-language-models-in-context-learning-0207.
Zheng, Lianmin, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, et al. 2023. “Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena.” arXiv. https://doi.org/10.48550/arXiv.2306.05685.
Ziegler, S. L. 2019. “Digitization Selection Criteria as Anti-Racist Action.” *The Code4Lib Journal*, no. 45. https://journal.code4lib.org/articles/14667.