Towards Scalable Structured Data from Clinical Text

by

Monica Agrawal

B.S., Stanford University (2017)
M.S., Stanford University (2018)

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2023

© Massachusetts Institute of Technology 2023. All rights reserved.

Author . . . .
Department of Electrical Engineering and Computer Science
January 25, 2023

Certified by . . . .
David Sontag
Professor of Electrical Engineering and Computer Science
Thesis Supervisor

Accepted by . . . .
Leslie A. Kolodziejski
Professor of Electrical Engineering and Computer Science
Chair, Department Committee for Graduate Students

Towards Scalable Structured Data from Clinical Text

by

Monica Agrawal

Submitted to the Department of Electrical Engineering and Computer Science
on January 25, 2023, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Abstract

The adoption of electronic health records (EHRs) presents an incredible opportunity to improve medicine both at the point-of-care and through retrospective research. Unfortunately, many pertinent variables are trapped in unstructured clinical note text. Automated extraction is difficult since clinical notes are written in their own jargon-heavy dialect, patient histories can contain hundreds of notes, and there is often minimal labeled data. In this thesis, I tackle these barriers from three interconnected angles: (i) the design of human-AI teams to speed up annotation workflows, (ii) the development of label-efficient modeling methods, and (iii) a re-design of electronic health records that incentivizes cleaner data at time of creation.

Thesis Supervisor: David Sontag
Title: Professor of Electrical Engineering and Computer Science

Acknowledgments

I would first like to thank my advisor, David Sontag, who has been nothing short of a force of nature during my PhD. David always encouraged me to approach research by dreaming bigger and sketching out a vision for what healthcare could be, without getting lost in the minutiae. At the same time, he has never shied from getting deep into the messy details of methods or data with me (and the data was often very messy). David has also taught me the value of discussing research over ice cream. I aspire to be able to emulate these philosophies with my own future mentees.

I would also like to thank the rest of my thesis committee for their contributions to my research development. Through class lectures, reading groups, and research conversations over the years, Peter Szolovits has helped me contextualize my research within machine learning and healthcare. I would also like to thank the members of his group who welcomed me into clinical NLP and acted as early sounding boards for my ideas. I first met with Noemie Elhadad during my first semester of the PhD; since then, she has welcomed me into the informatics community, served as a research mentor, and inspired me to work on research problems that others are not. Finally, I would like to thank Yoon Kim who has helped me navigate and probe the surprises of ever larger language models for the past four years. He has served as an invaluable co-author on several papers throughout my PhD, and I always looked forward to what would emerge from our brainstorming meetings.

I would have not gotten through my PhD without a plethora of mentors and collaborators. First, I would like to thank my collaborators at Harvard Medical (Steven Horng, Chloe O'Connell, Matthew Alkaitis, Nicholas Kurtzman, Irbaz Riaz, Barbara Lam) and Memorial Sloan Kettering (Lior Gazit, Pedram Razavi, Gregory Riely, Michael Offin) for giving me critical insight into clinical workflows and serving as indispensable co-authors on many of the papers that comprise this thesis. I would also like to thank David Karger and Arvind Satyanarayan; both of them profoundly influenced the human-centered lens through which I now view machine learning.

Next, I would also like to acknowledge the students who I co-authored with throughout the PhD, who made research a joy. I would like to thank Irene Chen, who ushered me through my first paper in the group and joined me in below freezing weather for the grand opening of a cookie store; Hunter Lang, who taught me equally about weak supervision and how to pull a successful all-nighter; Alexander Lew, who provided me guidance on both probabilistic programming and the best chicken sandwiches around; Luke Murray, who signed on to the daunting task of transforming electronic health records; and Stefan Hegselmann, who made a long conference road trip pass quickly. I would like to especially thank the students I mentored—Divya Gopinath, Ariel Levy, Jason Zhao, and Sharon Jiang; they have all taught me so much in return.

I would also like to thank all the labmates who overlapped with me in the Clinical Machine Learning group, especially Rahul Krishnan, Irene Chen, Michael Oberst, Rebecca Boiarsky, Zeshan Hussain, Christina Ji, Chandler Squires, Hussein Mozannar, Hunter Lang, Shannon Shen, Ilker Demirel, Stefan Hegselmann, Nikolaj Thams, Edward de Brouwer, Alejandro Buendia, Elizabeth Bondi-Kelly, Fredrik Johansson, Ming-Chieh Shih, and Mercy Asiedu. I have such fond memories of brainstorming on the couches, eating cardamom buns at fika, rooting for Federer and Nadal, hiking in too-cold weather, and reducing stress via trips to Dunkin (thank you Rebecca). I can't wait for all the conference reunions to come.

I would like to thank the mentors before MIT who were so crucial to my development. At Kittredge and Chamblee, I am so grateful to have had too many devoted teachers to name; they went above and beyond to support my endeavors both inside and outside of the classroom. At Stanford, I would like to thank Audrey Ellerbee Bowden, Gennifer Smith, Jure Leskovec, and Marinka Zitnik, for providing me the research experiences at Stanford that convinced me I wanted to pursue research further. I would also like to thank Keith Schwarz and Julie Zelenski, who both emailed me encouraging me to major in computer science, and Dan Jurafsky, who taught me one can never use too many exclamation points. I would also like to thank the mentors I had at Flatiron Health (Ben Birnbaum, Kathi Seidl-Rathkopf, Lucy He), who first showed me how clinical text data can change medicine.

I would also like to thank my friends outside academia who serve as constant reminders of world outside of paper deadlines. From Atlanta, I'd like to thank Kailyn LaPorte, Jessica Simon, and Krishna Inmula, who have stuck with me through it all, including my deep lack of fashion sense in middle school. From Stanford, I would like to thank Nathalia Scrimshaw and Priya Ganesan, who made Seattle an escape from graduate school for me, and Anshul Samar, who always reassured me I had a future in comedy if the PhD didn't work out (even though I truly don't).

I would like to thank GW6, for providing me a departmental community throughout my PhD, and Janet Fischer and Leslie Kolodziejski for facilitating this space. I would like to especially acknowledge Serena Booth, Sarah Cen, Lucy Chai, Aspen Hopkins, Irene Kuang, and Divya Shanmugam, who have provided me an avenue for celebration and commiseration during the PhD. In addition, Irene Kuang always seemed to make me butternut squash macaroni and cheese whenever I needed it the most, and I am so, so grateful.

I would also like to thank the many roommates who put up with me during the PhD: C. Wong, Max Vilgalys, George Fei, Wen Wen Teh, Jamie Gracie, and Alena Rote. The first three in particular lured me to join them in Cambridge from California with the promise of elaborate meals and freshly baked pastries; they more than made good on that promise. I would additionally like to thank the other friends who made my time in Cambridge so delightful, including Ankit Gupta, Sarah Young, and Joy Zeng.

I would like to extend my deepest gratitude to Patrick, who has been the most fervent supporter of (what he has named) the Monica Institute of Technology. In addition to his other innumerable positive qualities, Patrick has helped me parse measure theory proofs before I present in reading group, stayed up late to keep me company before my conference deadlines, and ensured I consumed fruit leading up to aforementioned deadlines.

I would like to thank my parents and brother who have been along for this journey for far longer than five years. I would like to thank my brother for walking me through my first program (even though it involved 1-indexing), my mother for her unwavering belief that I could do whatever I put my mind to, and my father for supporting my scientific research career from the very beginning. In particular, in 5th grade biology, I was too scared to dissect a frog on my own, and he left work to come help me. I am both truly sorry that happened and forever grateful. I can't thank the three of you enough.

Finally, I would like to acknowledge the sources of funding that made this dissertation possible: the Tau Beta Pi Fellowship, the MIT EECS Harold E. Edgerton '27 Fund Fellowship, the Takeda Fellowship, Memorial Sloan Kettering Cancer Center, the MIT Deshpande Center, and the MLA@CSAIL Initiatives.

This doctoral thesis has been examined by a Committee of the
Department of Electrical Engineering and Computer Science as follows:

Professor David Sontag . . . .
Thesis Superviser
Professor of Electrical Engineering and Computer Science, MIT

Professor Peter Szolovits . . . .
Member, Thesis Committee
Professor of Electrical Engineering and Computer Science, MIT

Professor Noemie Elhadad . . . .
Member, Thesis Committee
Associate Professor of Biomedical Informatics, Columbia

Professor Yoon Kim . . . .
Member, Thesis Committee
Assistant Professor of Electrical Engineering and Computer Science, MIT

Contents

1 Introduction 35

1.1 Promise of Electronic Health Records 35

1.2 Challenges 35

1.3 Contributions 38

2 Benchmarking Clinical Entity Extraction 41

2.1 Introduction 41

2.2 Related Work 43

2.3 Current Performance of Entity Normalization Systems 45

2.3.1 Data 45

2.3.2 Analysis Methodology 46

2.3.3 Analysis Results 47

2.4 Annotation Framework 48

2.4.1 Framework Development 49

2.4.2 Amended Framework 49

2.4.3 Comparison to MCN 52

2.5 Validation of Framework 52

2.5.1 Annotating under our Framework 52

2.5.2 Agreement Evaluation 54

2.5.3 Evaluation of Real World End-to-End Systems 54

2.6 Discussion 56

2.7 Conclusion 58

3 Decision Support Tool for Clinical Entity Annotation 59

3.1 Introduction 59
3.2 Related Work 60
3.3 System 61

4 Assessing the Impact of Decision Support 67

4.1 Inroduction 67
4.2 Related Work 69
  4.2.1 Mental Models and Model Trust 69
  4.2.2 User Agency 71
4.3 Annotation 71
4.4 Stage 1: Label Recommendations 72
  4.4.1 Experimental Design 72
  4.4.2 Results 75
4.5 Stage 2: Annotation Suggestions 79
  4.5.1 Experimental Design 79
  4.5.2 Results 81
4.6 Discussion 85
4.7 Conclusions 88

5 Human-AI Framework for Clinical Timeline Creation 91

5.1 Introduction 91
5.2 Related Work 92
5.3 Methods 95
  5.3.1 Preliminaries 95
  5.3.2 Event Extraction Model 95
  5.3.3 Human-guided extraction 96
5.4 Data 97
  5.4.1 Cohort 97
  5.4.2 Metastatic Recurrence 98
  5.4.3 Therapy Start 98

5.5 Experiments . . . 99
5.5.1 Comparisons . . . 100
5.5.2 Metastatic Recurrence . . . 101
5.5.3 Therapy Start . . . 104
5.6 Discussion . . . 106
5.7 Conclusions . . . 109

6 Self-supervision for Longitudinal Clinical Text 111
6.1 Introduction . . . 111
6.2 Related work . . . 114
6.3 Order-pretraining algorithm . . . 115
6.4 Finite-sample guarantee for time-irreversible features . . . 117
6.4.1 Extraction example . . . 119
6.5 Experiments . . . 122
6.5.1 Synthetic data . . . 122
6.5.2 Real-world data . . . 123
6.6 Limitations and Conclusion . . . 127

7 Large Language Models for Clinical Information Extraction 129
7.1 Introduction . . . 129
7.2 Related Work . . . 132
7.2.1 Prompt-Based Learning . . . 132
7.2.2 Pretrained LMs for Clinical NLP . . . 133
7.3 Methods . . . 134
7.3.1 Predicting Structured Outputs with LLMs . . . 134
7.3.2 Large Language Models Used . . . 135
7.3.3 Dataset Annotation . . . 136
7.4 Clinical Sense Disambiguation . . . 137
7.5 Biomedical Evidence Extraction . . . 139
7.6 Coreference Resolution . . . 141
7.7 Medication Extraction . . . 142

7.7.1 Recognition + Status Classification 143
7.7.2 Recognition + Relation Extraction 145
7.8 Analysis of Common Crawl Data 146
7.9 Conclusion 150
7.9.1 Limitations 152
7.9.2 Ethical Considerations 153

8 Introducing Autocomplete for Cleaner Electronic Health Records 155
8.1 Introduction 155
8.1.1 Contributions 156
8.2 Background: The Current State of Clinical Workflows 157
8.3 Methods 159
8.3.1 Data Overview and Cohort Definitions 159
8.3.2 Defining Labels for Autocompletion 159
8.3.3 Developing Predictive Features for Autocompletion 161
8.3.4 Autocompletion Models 162
8.4 Results 169
8.4.1 Performance and Usability Metrics 169
8.4.2 Autocompletion Sensitivity Analysis 172
8.4.3 Interpreting Contextual Autocompletion of Prior Conditions . 173
8.5 Discussion 175
8.6 Conclusion 177

9 Integrating Automated Information Retrieval into EHRs 179
9.1 Introduction 179
9.1.1 Contribution 180
9.2 Related Work 184
9.2.1 Information Capture 184
9.2.2 Information Fragmentation in EHRs 186
9.2.3 Problem-oriented Medical Records 187
9.2.4 Information Foraging Theory 188

9.3 Environment 189
  9.3.1 Clinical Workflow 189
  9.3.2 A Variety of Documentation Processes 190
  9.3.3 Study Environment 191
9.4 Design and Implementation 191
  9.4.1 Autocomplete 192
  9.4.2 Post Recognitions 194
  9.4.3 Semantic Highlighting and Concept Disambiguation 195
  9.4.4 Context-specific information retrieval 197
  9.4.5 Default Text 198
  9.4.6 Concept-Oriented Views 199
  9.4.7 Surfacing Cards 200
  9.4.8 Hand Designed vs. Automatically Generated Cards 201
  9.4.9 Card Design 202
  9.4.10 Inline Display of Cards 204
9.5 Evaluation 204
9.6 Discussion 209
9.7 Conclusion 212

10 Conclusion **215**

A Additional Information for Chapter 2 **221**
A.1 Description of Annotation Tool Suggestions 221
A.2 Performance of Systems by Semantic Type 222
A.3 Full Clinical Annotation Schema Guidelines 223

B Additional Information for Chapter 5 **229**
B.1 Consort Diagrams 229
B.2 Proof of Reduction to Binary Search 231
B.3 Reinforcement Learning Details 232
  B.3.1 Decision-making agent 232

B.3.2 Reinforcement learning 233

C Additional Information for Chapter 6 **235**

C.1 Synthetic Experiments 235

C.1.1 Synthetic Distributions 235

C.1.2 Experimental Setup 236

C.1.3 Explanation of Observed Behavior 237

C.2 Real-world Experiments 239

C.2.1 Pre-training for feature selection 239

C.2.2 Nonlinear representations 241

D Additional Information for Chapter 7 **245**

D.1 Prompts and Sample GPT-3 Outputs 245

D.1.1 Clinical Sense Disambiguation 246

D.1.2 Biomedical Evidence Extraction 247

D.1.3 Coreference Resolution 249

D.1.4 Medication Status Extraction 251

D.1.5 Medication Attribute Extraction 253

D.2 Annotation Process 256

D.2.1 Biomedical Evidence Extraction 256

D.2.2 Coreference Resolution 256

D.2.3 Medication Status Extraction 257

D.2.4 Medication Attribute Extraction 257

D.2.5 Clinical Sense Disambiguation 259

D.2.6 Biomedical Evidence Extraction 259

D.2.7 Coreference Resolution 262

D.2.8 Medication + Status Extraction 262

D.2.9 Medication + Relation Extraction 264

D.3 Experimental Cost 266

D.4 Website Category Classification 267

D.4.1 Accuracy 267

D.4.2 Examples 268

E Additional Information for Chapter 8 **271**

E.1 Data Extraction and Featurization 271

E.1.1 Examples of Clinical Notes 271

E.1.2 The NegEx Algorithm 273

E.1.3 Trie-Based Extraction of UMLS Concepts 273

E.1.4 Bucketization of Triage Vitals 276

E.2 Extended Autocomplete Performance 277

E.2.1 Autocompletion Scope and Type Detection 277

E.2.2 Training Contextual Model for Conditions 278

E.2.3 Retrospective Autocompletion Performance using MRR, MAP, and Keystroke Burden 279

E.2.4 Ontologies and Code 280

List of Figures

1-1 Diagram indicating sample variables needed to structure longitudinal clinical data, and the difficulty of scaling. 36

1-2 Example of the messiness of raw clinical text, and its transformation to a structured format. 37

2-1 Pipeline of extracting UMLS concepts from text. Typically, this is broken down into two steps as shown: recognition (detecting relevant spans) and normalization (mapping spans to a vocabulary). 42

2-2 Examples of compound concepts represented inconsistently in SNOMED. While there is a term for 'Severe Asthma Exacerbation', there is no term for 'Severe COPD Exacerbation.' Existing datasets only tag the longest concept available in the vocabulary, but this makes it difficult to annotate concepts consistently. 51

2-3 Custom in-house annotation tool, an early prototype of PRAnCER described in Chapter 3. Annotators highlight the span they want to tag in the middle panel (pictured: cold), and the panel on the left automatically provides suggestions for CUIs, which can be scrolled through and selected. If the suggestions provided are insufficient, an annotator can use the UMLS browser and directly enter CUIs instead. The right panel shows the labels that were already tagged and at which character locations: the pencil allows for edits and the trash can for deletions. 53

3-1 Our clinical annotation system in the midst of annotating our sample sentence. Panel a) shows the text, label, and selection panels. In the above, *Pt* has already been annotated; the user annotates *carbo* using the search feature in panels a) and b), *ia* using the automatically surfaced label recommendations in panel c), and *TNBC* using a pre-populated annotation suggestion in panel d). 61

3-2 Tutorial mode built into the platform to train users with rounds of iterative feedback. Gold standard and chosen annotations are displayed side-by-side, with feedback on all annotations provided below. 64

4-1 Accuracy (total recall) and efficiency (time to label) results for users with label recommendations (*Standard* and *Weakened* modes) and users without (*None* mode). 74

4-2 Flag indicators for suggestion confidence, displayed in the selection panel. Low-confidence (left) is indicated by an exclamation point on a red background, and high-confidence (right) is indicated by a star. . . 79

5-1 Plots of the model's cumulative probability over time that a given patient was metastatic. For patients 1 and 2 on the left, due to the sudden jump in cumulative probability, the model feels sufficiently confident to directly extract the date without eliciting any human input. For patient 3, on the right, before any queries are made, there is more ambiguity as to the date of metastatic recurrence. However, after a single query at the marked index, the model calculates a new posterior probability and is now sufficiently confident to extract. 93

5-2 The above plot visualizes the trade-off between the average number of queries per patient, and the percentage of patients whose estimated date of metastatic recurrence fell within one month of the first metastatic note. At the bottom left, one can see the accuracy if the extraction model was used alone, and on the top right, the number of queries necessary if binary search was used to pinpoint each variable. We can observe that employment of *Model-derived Query Utility* provides the optimal trade-off, compared to the other methods. 102

5-3 The left plot shows the distribution of number of queries solicited under the *Model-derived Query Utility* algorithm at L=1, in which the correct date was pinpointed within a month 98% of the time; 57% of patients required no queries at all. The right plot shows the distribution of initial errors of the extraction model, split into (i) the patients whose dates were directly extracted and (ii) the patients for whom queries were requested. We note that there were minimal errors on the set not queried. 103

5-4 Results for the medication extraction task. The plot above shows the trade-off between the average number of queries per patient, and the percentage of patients that fall within one month of the gold standard therapy start dates. *Model-augmented binary search* and *Model-derived Query Utility* perform comparably and provide the most consistent performance across hyperparameter choices. 105

6-1 Depiction of the data-generation and learning process for order-contrastive pre-training. For each trajectory, a pair of consecutive windows is sampled uniformly at random, flipped with probability 0.5, and presented to the model. The model is trained to predict whether the presented pair of windows is in the correct (+1) or incorrect (−1) order. 113

6-2 Synthetic experiments admitting different behavior across pre-training setups. (Top): PCL is unable to ever recover all four true features in S; (Bottom): PCL recovers the true representation, but requires higher sample complexity than OCP. 122

6-3 Linear representation space experiment to validate assumptions of our model apply to real-world data. Quantitatively, we find downstream wins from restricting the model feature space to those found useful for the order-contrastive task. Qualitatively, the features important for the order pre-training are the same we would expect to be useful for the downstream extraction task. 124

7-1 Illustration of our approach using a one-shot example (green) and guidance (brown) to create a more structured LM output (blue). This significantly reduces the necessary post-processing effort of a resolver (gray). 130

8-1 Semi-structuring notes with contextual autocompletion can enable extensive changes to both documentation and clinical decision support. 157

8-2 Data model for our ontology of conditions. Clinical notes are normalized to UMLS, and sets of UMLS IDs (CUIs) are aggregated to create unique concepts in our ontology. Ontology entries are then grouped together in coarser model relevancy buckets. 159

8-3 Screenshots of contextual autocompletion tool for each autocompletion type. From left to right: (a) Conditions (b) Symptoms (c) Medications and (d) Labs. Trigger words before the tagged term affect the scope and type of the autocompletion. Clinical concepts with synonyms that match the typed text are listed with the synonym in black text and the more general concept name in gray. 168

8-4 Case Studies of Rankings over Conditions 175

9-1 The MedKnowts interface containing sections of the clinical note on the left, and an integrated sidebar on the right. The user is typing `WBC` and triggering autocomplete (A). Detail text in the autocomplete is used to differentiate clinical terms and provide additional context such as result counts. The card for the most recent identified term, `CK`, is displayed in the preview pane (C) with values displayed as a line chart, and abnormal values highlighted in red. The preview pane history can be navigated using the backwards and forwards buttons at the top of (C). Below the preview pane the doctor has pinned a card for `oxycodone` (D), which displays note snippets relevant to `oxycodone`. A search bar at the top of the sidebar can be used as an alternative method to add cards to the preview pane. A transcluded card for cardiac conditions (B) shows labs, cardiology reports, and note snippets relevant to congestive heart failure in a single interface. 181

9-2 Autocomplete in the same context without filters and with filters. The "/m" command is used to limit the clinical terms displayed to medications. 193

9-3 Autocomplete inserts terms as highlighted immutable chips. They can be deleted, moved around and copied like other text, but they cannot be modified. 195

9-4 Post recognitions are automatically recognized clinical terms. They are rendered with a dotted border and can be disambiguated through a popup menu on click. Negated post recognitions are rendered with an underline. 196

9-5 An example of context-specific information retrieval. Autocomplete insertion of lab results using a tree based menu with support for aggregation at multiple time frames and specific values 197

9-6 An example of the review of systems section 198

9-7 An example card surfaced for *Congestive Heart Failure*, which contains pertinent lab values, links to recent echocardiography reports, and recent notes that mention the condition. 203

9-8 Scribes' System Usability Scale scores with medians displayed as black bars for each question 206

D-1 Platform used for annotation of the three new labeled datasets, shown for the coreference resolution annotation task. 256

E-1 Pseudocode of the rule-based negation detection algorithm. 274
E-2 Screenshots of our backup data capture strategies in the case that autocompletion scope detection algorithms fail. (a) Users can insert a slash character (`/`), which acts as a manual trigger to force autocompletion. (b) Users can retroactively accept tags for candidate concepts that they typed but did not autocomplete. 278
E-3 Mean MRR for five conditions (± 95% CI from mean) using contextual and frequency-based autocompletion. Concepts were chosen to get representative samples of the data. 281

List of Tables

2.1 Performance of the top 10 systems on the entity normalization task on the MCN corpus. Accuracy is measured on different subsets of the test data and by the 3 metrics described in Evaluation Methodology . . . 47

2.2 The three significant changes in our annotation schema, in response to the main limitations we found in current annotation frameworks. . . 49

2.3 Performance of MetaMap and cTAKES on the whole dataset created under our framework, and subdivided by the ten most common Semantic Types. *% Spans Correct* indicates what percent of spans were correctly recognized by the model, *% CUIs correct* indicate what percent of spans were correctly recognized with a correct CUI, and *CUI precision* indicates the percent of correctly identified spans that were normalized correctly. 55

4.1 The percentage of times users chose to initiate a search. Across modes, searches are rarely initiated when the true label is provided in the recommendations (“Easy"), and are often initiated when the true label isn’t provided (“Difficult"). When a user expects a label to be provided but it is not (*Weakened* mode on “Weakened" examples), users search less (73%) than they did when they didn’t expect the label to show up (82%). 78

4.2 Differences for four representative users between nontrivial span recall on Stage 1 (where they had no pre-annotations) and Stage 2 (where they did). Nontrivial span recall calculates the proportion of spans that users took the initiative to annotate that would not have been pre-annotated by our annotation suggestion algorithm. While the drop between Stage 1 and Stage 2 recall indicates that users took less initiative in practice, users believed they were more thorough in their exit surveys. 83

6.1 Performance of deep methods on cancer progression extraction. The first row contains the mean AUC of OCP ± its std dev. The following rows contain the mean AUC advantage of OCP over each comparison method, and the percentage of time OCP outperforms that method, across the 3 seeds and 5 folds. 125

7.1 Overview of the five tasks studied in this chapter and the datasets that were used. 131

7.2 **Clinical sense disambiguation.** Accuracy and macro F1 for zero-shot language modeling approaches on a subset of the Clinical Acronym Sense Inventory (CASI) data set [176] and the MIMIC Reverse substitution dataset [1]. GPT-3 is not run on MIMIC due to the data use agreement. To evaluate on MIMIC we distill GPT-3 + R into a smaller model by treating the outputs as weak supervision and following Lang et al. [144] "+ distillation", then evaluate the smaller model on MIMIC as well. . 138

7.3 **Biomedical Evidence Extraction**. Test F1 scores on the binary token-level sequence labeling problem for Intervention identification [196], and abstract-level accuracy at study arm identification. The supervised baselines were trained on 4,800 abstracts. 140

7.4 **Coreference Resolution**. Macro unigram recall and unigram precision of methods on our newly annotated task using CASI [176]. The end-to-end baseline was trained on three non-clinical coreference resolution datasets and transferred to this new setting. 1-shot results are averaged over 5 prompts. 142

7.5 **Medication extraction.** Micro recall and precision for medication extraction on our self-annotated dataset. 143

7.6 **Medication status classification.** Conditional accuracy and macro F1-score for Identification of medication status *active*, *discontinued*, and *neither*. 143

7.7 **Medication attribute extraction.** F1 scores on our newly annotated medication extraction dataset. The baselines are trained using supervised learning on i2b2 [276], then transferred to the test domain. *Relation Extraction* additionally requires the model to match modifiers (dosage, route, etc.) to the medication span. Baseline end-to-end relation extraction performance suffers due to errors cascading from the extraction step. 144

7.8 Number of search results for a subset of the clinical text pairs, as ascertained from the AI2 utility for searching Colossal Clean Crawled Corpus (C4). 149

7.9 Distribution of the sources of data in which clinical text pairs are found, as classified by `text-davinci-003`. *CASI median* refers to a set of 100 pairs derived from the CASI dataset, and *Colloquial Median* refers to a set of 10 pairs of common, manually created clinical text jargon. *For maximum observed, we exclude query pairs with fewer than 15 results, to prevent signal from a noisy handful of examples. 151

8.1 Retrospective Evaluation of MRR using Contextual Autocompletion. We report average MRR (±95% confidence interval of the mean) for each of our learned contextual autocomplete models, and compare our best models (dual-branched neural network for conditions, empirical conditioning on the chief complaint and most abnormal vital for symptoms) to spell-based and frequency-based baselines, both for specific concept types as well as overall using our scope and type prediction algorithms. Calculated across 25,000 visits. 171

8.2 Live Evaluation of Contextual Autocompletion Models. Mean keystroke burden for autocompleted concepts (± 95% CI from mean), measured across 40 notes written live by a single physician over two shifts. Performance is also broken down by note section, as well as concept type. 172

8.3 Number of keystrokes saved by our contextual model compared to a frequency-based baseline (±95% CI of the mean) for conditions. Performance was stratified by concept frequency (by terciles) and by available medical history. 173

8.4 Predictive features for selected condition concepts, using a linear approximation to our contextual model for conditions. Inputs to the model are a TF-IDF representation of the triage text as well as the presence of coarse-grained model relevancy buckets in a patient's prior medical record, as defined in Section 8.3.2. 173

9.1 General usage data. Totals for Patients and Shifts are left out because the scribes worked with the physician on the same patient/shift. Pins reflect the number of cards each user pinned to the sidebar. 205

9.2 How users inserted chips. Autocomplete indicates insertion via autocomplete and post recognition is disambiguation of post recognitions. 205

9.3 How users added cards to the sidebar. Autocomplete means clicking on an autocomplete chip, post recognition means clicking on a post recognition, search means performing a search in the sidebar, and note snippet means clicking on a note snippet in a card. Clicking on a note snippet displays a card containing the full note text with the snippet highlighted. 206

C.1 AUC ± standard deviations (as calculated over five folds), comparing the use of all features to just those selected by OCP. The final row displays the difference in AUC between OCP-selected features and all features and the standard deviation of that difference. 239

C.2 Average precision at 80% recall ± standard deviations (as calculated over five folds), comparing the use of all features to just those selected by OCP. The final row displays the difference in average precision between OCP-selected features and all features. 240

C.3 AUC average and standard deviation (as calculated over five folds) over each of the three seeds trained per method. 242

C.4 Performance in terms of average precision at 80% recall as calculated across 5 folds and 3 seeds for fine-tuned methods. The first row contains the mean average precision of OCP ± its standard deviation. The following rows contain the mean precision advantage of OCP over each comparison method, and the percentage of time OCP outperforms that method, computed over all seeds and folds. 243

D.1 Estimate of cost of running the experiments included in this chapter . 266

D.2 Number of search results in each category of the 8608 scraped documents, as classified by `text-davinci-003`. 267

D.3 Examples for the eight major categories (excluding other/non-health) of websites containing medical jargon. Examples shown for carbo/carboplatin, except for the `Medical Encyclopedia/Dictionary` option. . 269

E.1 Comparing NER approaches on OMR notes 276

E.2 Retrospective Evaluation of Keystroke Burden and MAP using Contextual Autocompletion. We report the mean keystroke burden/MAP for the contextual autocomplete models we prototyped for each concept type, following the conventions of Figure 8.1 280

Glossary of Recurring Acronyms

- AI: artificial intelligence
- API: application programming interface
- AUC: area under the receiver operating characteristic curve (also AUROC)
- BERT: Bidirectional Encoder Representations from Transformers
- BIO: beginning-inside-outside annotation model
- BOW: bag-of-words
- CASI: Clinical Acronym Sense Inventory
- CRF: conditional random field
- cTAKES: clinical Text Analysis and Knowledge Extraction System
- CUI: Concept Unique Identifier
- EBM: evidence-based medicine
- ED: emergency department
- EHR: electronic health record
- GPT: generative pre-trained transformer
- GPT-3 + R: Resolved GPT-3 approach (introduced by thesis)
- HPI: History of Present Illness section
- i2b2: Informatics for Integrating Biology & the Bedside
- ICA: independent component analysis
- IRB: institutional review board
- L_0 regularization: regression with fixed number of non-zero parameters

- L_1 regularization: lasso regression
- L_2 regularization: ridge regression
- LASSO: least absolute shrinkage and selection operator
- LM: language model
- LLM: large language model
- LOC: lines of code
- LSTM: Long Short-term Memory
- ML: machine learning
- MRR: mean reciprocal rank
- MAP: mean average precision
- MDM: Medical Decision Making section
- MDP: Markov decision process
- MIMIC: Medical Information Mart for Intensive Care
- MCN: Medical Concept Normalization corpus
- n2c2: National NLP Clinical Challenges
- NER: named entity recognition
- NLP: natural language processing
- OCP: order contrastive pre-training (introduced by thesis)
- PCL: permutation contrastive pre-training
- POMR: problem-oriented medical record

- PRAnCER: Platform enabling Rapid Annotation for Clinical Entity Recognition (introduced by thesis)
- RCT: randomized controlled trial
- ROS: Review of Systems section
- RWE: real-world evidence
- SNOMED CT: SNOMED (organization) Clinical Terms
- TF-IDF: term frequency-inverse document frequency
- UI: user interface
- UMLS: United Medical Language System

Chapter 1

Introduction

1.1 Promise of Electronic Health Records

The widespread adoption of electronic health records (EHRs) has made it increasingly feasible to use data from routine clinical practice to better understand the effectiveness of treatments in real-world settings [246, 227, 78, 129]. Real-world evidence (RWE) can help address a host of issues that cannot be fully studied in randomized controlled trials (RCTs), such as generalizability of clinical practice across settings, heteregeneity of outcomes across subpopulations, and postmarketing surveillance [78, 69, 76, 43]. Such observational studies leveraging RWE are particularly important in diseases where the treatment landscapes have evolved rapidly [104, 109]. For example, in renal cell carcinoma (RCC), ten new targeted drugs have been approved since 2006 [17]. In addition to its potential to transform retrospective research, the adoption of EHRs could also transform medicine by providing clinicial decision support at the point-of-care, enabling the discovery of new clinically meaningful phenotypes, and streamlining largely manual workflows, e.g., clinical trial matching [118, 185, 60].

1.2 Challenges

Unfortunately, this promise is far from fully realized. Answering such questions requires many variables—from side effects to the outcomes themselves—that cannot

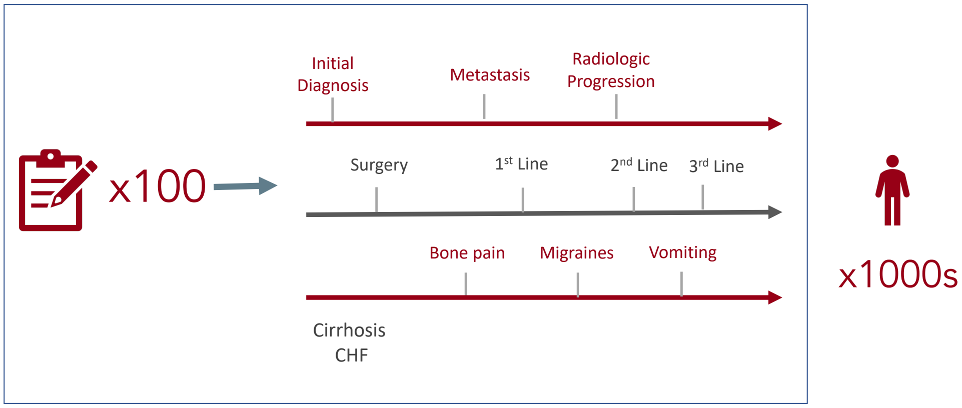
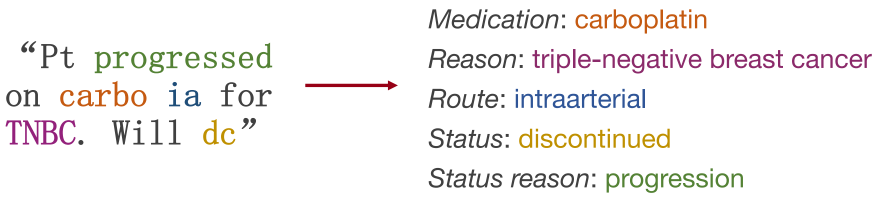


Figure 1-1: Diagram indicating sample variables needed to structure longitudinal clinical data, and the difficulty of scaling.

be found within the structured data in electronic health records. Even when data is seemingly present in structured form, it is often not accurate or well-maintained. For example, medication lists are often not updated to reflect when a patient stops taking a drug [34], and problem lists face similar issues [280]. Instead, they are trapped within unstructured clinical notes; due to its flexibility, free text is preferred by clinicians for efficient data entry. Clinical notes are an incredibly rich source of data about a patient's interactions with the healthcare system. They are lengthy, containing detailed information about a patient's state, their underlying conditions, and any procedures performed.

As a result, in order to conduct the kinds of retrospective research studies previously described, the status quo is to undergo a manual *chart review* process [12, 29]. In chart review, clinical domain experts manually transform messy raw clinical notes into a structured format that can then be ingested, e.g., by comparative effectiveness studies. Necessary variables include disease status over time, intervention plan over time, adverse events, and comorbidities, as shown in Figure 1-1. However, this can be incredibly time-consuming, as patients can have on the order of hundreds of notes, and this process needs to be conducted for large cohorts of patients to draw meaningful clinical conclusions. This manual process is prohibitively slow, limiting the scope of questions researchers can explore. Additionally, the lack of reliable structured data is also harmful for clinicians at the point-of-care, who have to exert extra effort sifting



“Pt progressed on carbo ia for TNBC. Will dc”

Medication: carboplatin
Reason: triple-negative breast cancer
Route: intraarterial
Status: discontinued
Status reason: progression

Figure 1-2: Example of the messiness of raw clinical text, and its transformation to a structured format.

through a patient’s written record, in order to understand the clinical context required for patient care.

Therefore, the goal of the long-standing field of clinical information extraction is to automate the transformation of unstructured clinical text to a useful structured form [288]. Unfortunately, there are many challenges involved. First, clinical notes often consist of overloaded terminology and frequent use of shorthand, which often differs across medical specialties and institutions. As an example, Figure 1-2 shows the complexity required to structure even just a short snippet of text.

As a result of the large amount of jargon in clinical text and the difficulty of annotation, there is often only a minimal amount of labeled data available to act as training data for natural language processing extraction models. This is further compounded by the difficulty of sharing clinical text annotations across institutions, due to the sensitivity of clinical text. Therefore, the large-scale annotation needed for machine learning algorithms can be prohibitively expensive. Due to the lack of sufficient training data, many existing approaches in clinical information extraction have relied on hand-written rules, which are task-specific, time-consuming to construct, and often brittle [65, 136]. Neither of these approaches—time-consuming annotation or rule-based approaches—are built to scale and generalize across clinical use cases (since rare medical concepts are generally insufficiently observed outside of large-scale datasets) and settings (due to the diversity of language across departments and institutions).

1.3 Contributions

In this thesis, I present machine learning and natural language processing solutions to enable the generation of structured health data at scale. In the chapters that follow, I address the aforementioned challenges by focusing on three broad themes: joint human-AI teams, label-efficient machine learning, and smarter electronic health records. I begin in Chapter 2 (originally published at MLHC 2020 as [4]) by analyzing the failure modes of existing clinical concept extraction methods to motivate the chapters that fellow:

Joint Human-AI Frameworks for Annotation: Manual chart review is incredibly tedious and a common bottleneck in clinical research and clinical NLP timelines. Consequently, I explore human-AI systems to optimize and streamline the efforts of expert annotators.

- Chapter 3 (published as a part of [150] at CHI 2021) introduces PRAnCER, an open-sourced tool for clinical entity annotation that enables the rapid annotation of medical terms to large ontologies within clinical notes. The decision support available in the tool can significantly speed up workflows and allow annotators to focus their time and energy on harder examples instead.

- Chapter 4 (originally published at CHI 2021 as [150]) assesses the impact of the decision support introduced in Chapter 3 on clinical annotators. While decision support can accelerate tedious tasks, a key concern is whether users overly trust or cede agency to automation. Through user studies, we find that 18 clinicians generally build intuition of when to rely on automation and when to exercise their own judgement. However, when presented with fully pre-populated suggestions, these expert users exhibit less agency. Our findings inform how systems and algorithms should be designed to mitigate the observed issues.

- Chapter 5 (originally published at MLHC 2021 as [324]) extends the idea of joint-human AI annotation to longitudinal notes and the structuring of clinical timelines. Natural language processing systems present an opportunity for automating the temporal identification of clinical events, but algorithms still have trouble accurately parsing the most complex patient cases. Chapter 5 presents a framework that automatically structures simple patient cases, but iteratively requests and directs human input when required. The framework is evaluated on identification of two event types in oncology.

Label-efficient algorithms: Ideally, we would reserve human effort and annotated data for robust model evaluation, rather than data-hungry training algorithms. With this in mind, I have developed novel machine learning methods requiring no labels (zero-shot) or few labels (few-shot) during training.

- Chapter 6 (published at AISTATS 2022 as [6]) explores an "order-contrastive" method for self-supervised pre-training on longitudinal data. We sample pairs of time segments, switch the order for half of them, and train a model to predict whether a given pair is in the correct order. Intuitively, the ordering task allows the model to attend to the *least time-reversible* features (for example, features that indicate progression of a chronic disease). We provide a principled intuition and demonstrate the empirical effectiveness of the self-supervised objective on a task identifying instances of cancer progression. These results indicate that pre-training methods *designed for* particular classes of distributions and downstream tasks can improve the performance of self-supervised learning.

- Chapter 7 (originally published at EMNLP 2022 as [5]) demonstrates that large language models (LLMs), such as InstructGPT [199], perform well at zero- and few-shot information extraction from clinical text despite not being trained specifically for the clinical domain. We show how to leverage LLMs to tackle a diverse set of NLP tasks which require more structured outputs, including span identification, token-level sequence classification, and relation extraction, and we introduce new datasets for benchmarking few-shot clinical information extraction.

Re-imagining electronic health records: Ultimately, it would be best if the clinical documentation process were cleaner to begin with and captured the variables we care about at the time of creation. However, augmenting the existing EHR with structured fields is not a viable solution, due to the additional time burden. Therefore, the last section of this thesis describes work augmenting EHR interfaces with machine learning to streamline the clinical documentation process, all while yielding cleaner data for researchers.

- Chapter 8 (originally published at MLHC 2020 as [90]) presents a system that uses a patient-specific autocompletion mechanism to facilitate rapid creation of semi-structured clinical documentation. We dynamically suggest relevant clinical concepts in real-time as a doctor drafts a note by leveraging features from both unstructured and structured medical data. To our knowledge, this system was the first machine learning-based documentation utility for clinical notes deployed in a live hospital setting.

- Chapter 9 (Originally published at UIST 2021 as [184]) rethinks navigation within electronic health records by enabling rapid contextual access to the patient's medical record. Chapter 9 introduces MedKnowts, an integrated note-taking editor and information retrieval system. Leveraging the semi-structured data achieved in the previous chapter, MedKnowts unifies the documentation and search process and provides concise synthesized concept-oriented slices of the patient's medical record. This enable easier parsing of long notes, auto-populated text, and proactive information retrieval, easing the documentation burden.

Chapter 2

Benchmarking Clinical Entity Extraction

Acknowledgement of Co-authors I would like to thank Chloe O'Connell for her lead on development of the annotation schema described in this chapter, Chloe and Yasmin Fatemi for the annotation of the dataset described in this work, and my undergraduate mentee Ariel Levy for her work on development of the annotation platform described.

2.1 Introduction

In this chapter, I begin by benchmarking the quality of existing algorithms in clinical entity extraction, to motivate the work that follows.

The strengths and weaknesses of clinical notes lie in the diversity of natural language. For example, `Cold` can refer to the temperature, a temperament, the viral infection, or Chronic Obstructive Lung Disease. By matter of preference, a doctor can refer to a patient with a 101 ° F temperature as `running a fever`, `being febrile`, or `having pyrexia`.

In order to transform text into a unified structured format useful for downstream applications, EHR text mining often involves recognizing spans representing concepts (named entity recognition) and mapping these spans to a common vocabulary (named

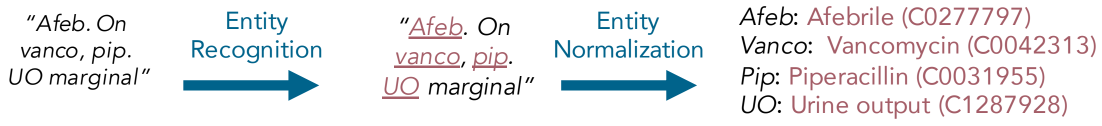

"Afeb. On vanco, pip. UO marginal"
Entity Recognition
"Afeb. On vanco, pip. UO marginal"
Entity Normalization
Afeb: Afebrile (C0277797)
Vanco: Vancomycin (C0042313)
Pip: Piperacillin (C0031955)
UO: Urine output (C1287928)

Figure 2-1: Pipeline of extracting UMLS concepts from text. Typically, this is broken down into two steps as shown: recognition (detecting relevant spans) and normalization (mapping spans to a vocabulary).

entity normalization/linkage). We will refer to both steps together as clinical entity extraction. An example of this two-step process is shown in Figure 2-1. Typically, in medicine, concepts are mapped to terms in the Unified Medical Language System (UMLS) or its subvocabularies, e.g., SNOMED CT [30, 194]. Each term is denoted by a Concept Unique Identifier (CUI).

Due to its utility, clinical entity extraction is widely used as an initial step in many machine learning for healthcare pipelines. For example, clinical entities can help construct cohorts, identify adverse drug events, spot potential confounders, learn a health knowledge graph, or match patients to relevant clinical trials [118, 228, 185, 60, 228]. They can also speed up manual, administrative tasks at hospitals, such as medical coding for billing and revenue cycle management.

Despite the importance of clinical entity extraction for a range of tasks, it still remains largely an open problem, and clinical research often has to resort to manual chart review for high-fidelity data, which is tedious and difficult to scale. While there has been a significant, concerted effort from the clinical natural language processing (NLP) community to bridge this gap, performance falters due to (i) the lack of sufficient annotated training data, (ii) the huge space of clinical entities to map to, and (iii) known issues with medical vocabularies.

Contributions

In this chapter, we investigate, quantify performance of, and indicate areas of improvements for the current state of normalization by analyzing performance on the

Medical Concept Normalization (MCN) Corpus [164]. The MCN corpus consists of 100 discharge summaries and 10,000 annotated concept mentions. They focus on a subset of CUIs in the UMLS: the SNOMED Clinical Terms, which consist of over 400,000 common clinical concepts; and RxNorm, a comprehensive clinical drug vocabulary [164, 194, 187]. On this corpus, we examine top systems for clinical entity normalization from the 2019 National NLP Clinical Challenges (n2c2) Shared Task as well as two widely used end-to-end open-source systems, cTAKES and MetaMap [15, 236]. These analyses reveal great heterogeneity in performance across various subsets of the data, for both normalization and end-to-end systems. This performance underscores the need for future dataset creation and method development in this space and highlights the need for a new evaluation framework. In their description of the MCN corpus, Luo et al. [164] recounted how their corpus annotation effort was hindered by several issues including imperfections of UMLS. Therefore, we additionally develop a new annotation framework to address these and other identified issues. We demonstrate how our new schema adjusts to these issues and allows for more flexible evaluation of end-to-end systems. We conclude by proposing recommendations to the clinical NLP community regarding a path forward for the creation of a reference standard, in hopes of spurring progress in the field.

2.2 Related Work

In the general domain, datasets for named entity recognition and normalization are huge, leading to the success of data-hungry algorithms. The Wiki-Disamb30 dataset, for example, consists of 1.4 million texts [72]. Techniques also often pull in external information: e.g., population priors, term-term co-occurrence, rich term metadata, and knowledge graphs [112]. Recent successful methods include end-to-end systems and DeepType, which uses a type classifier trained on 800 million tokens [216, 134]. However, these do not translate to the clinical domain because of the dearth of rich annotated clinical text. While a dataset of the magnitude seen in the general domain is infeasible in healthcare, one of our motivations for the creation of a larger dataset stems from successes in the general domain. Another drawback of the clinical domain is the comparative lack of entity information. UMLS contains a Related Concepts table with entity-entity relationships, including triples such as (‘Gematicin’, ‘is_a’, ‘antibiotic’). Unfortunately, these documented relationships are unreliable and incomplete [102, 50, 2, 53]. As a result, the table cannot be cleanly used to leverage entity-entity co-occurrences or other relationships, and more creative solutions are required.

Due to the importance of clinical entity extraction, there exist several open-sourced clinical entity extraction systems, such as cTAKES, MetaMap, and MedLEE [236, 15, 80]. These systems are very popular; over the past year, cTAKES has been mentioned in over 300 papers, MetaMap in over 500, and MedLEE in over 150. Many of these papers use these systems as a tool in their analysis. Unfortunately, past work has benchmarked the performance of these open-source systems in limited settings to middling results [219]; for example, Wu et al. [309] found that recall was below 0.5 for all systems on abbreviations specifically. However, due to the lack of a reference dataset, such benchmark numbers don’t exist over a diverse set of terms.

Several challenges in the clinical NLP community have focused on addressing entity recognition and/or normalization [47]. The 2010 i2b2/VA challenge released a dataset of 871 discharge summaries for recognition (but not normalization) of problems, tests, and treatments [277]. The 2012 i2b2 challenge released a similar dataset of 310 discharge summaries, that additionally annotated clinical departments, evidentials (e.g., ‘complained of’), and occurrences (e.g., ‘transfer’, ‘admission’). State-of-the-art for the clinical entity recognition task leverages contextual embeddings [249]. However, performance there is inconsistent, e.g., with an F1 score of 85.1 for treatment span detection, but only an F1 score of 66.3 for occurence detection.

Some clinical datasets also include span normalization. ShARe/Clef eHealth 2013 Task 1, SemEval-2014 Task 7, and SemEval-2015 Task 14 provide normalization to CUIs, but only for disease/disorder concepts [66, 212, 264]. Recently, Luo et al. [164] derived the Medical Concept Normalization Corpus (MCN) from the spans of 100 of the discharge summaries from the 2010 i2b2/VA challenge. They normalized over

13,600 entity spans (problems, tests, and treatments) to terms from SNOMED and RxNorm [164]. Track 3 of the 2019 National NLP Clinical Challenges (n2c2) Shared Tasks and Workshop utilized the MCN corpus. The challenge yielded 108 submissions, with a top submission accuracy of 85.2%.

In their paper, the creators of the MCN corpus detailed issues they faced in normalizing terms [164]. For example, if a compound term was not available in their vocabulary, the authors split the term into its constituents instead of labeling it as CUI-less. Since there is no term for `percutaneous drains`, they instead split the term into 'percutaneous (C0522523)' and 'drains (C0180499).' Since they used the spans from the 2010 i2b2 challenge, they recounted that spans were inconsistently annotated and were suboptimal for the normalization task. They also described inherent issues in the SNOMED vocabulary, such as inconsistency, duplication, and missing terms. These issues led them to have to make self-described arbitrary decisions in normalization. In this chapter, we build on the findings of Luo et al. [164] to address these issues in an annotation schema.

2.3 Current Performance of Entity Normalization Systems

2.3.1 Data

To understand the current state of clinical entity normalization, we analyzed the outcomes of Track 3 of the 2019 National NLP Clinical Challenges (n2c2) Shared Task, run on the MCN dataset [164]. Participants had to provide a single corresponding CUI for each provided span. In the MCN annotation process, if the original span from i2b2 (e.g., `high-grade fever`) was split up, teams were presented with the split spans (`high-grade` and `fever`), and not the original span. Teams trained on 50 tagged discharge summaries with 50 held out for evaluation. There were 108 total submissions, and here we analyze the outputs from the top 10 teams. Top teams used a variety of techniques ranging from classical NLP and information retrieval techniques to modern neural approaches, like contextual embeddings. To place performance in context of well-known systems, we compare to output from the Default Clinical Pipeline from cTAKES version 4.0.0 and the 2018 release of MetaMap [236, 15].

2.3.2 Analysis Methodology

We evaluate performance of the top 10 performing systems on the test data using 3 metrics: (i) highest accuracy any system achieved, (ii) average accuracy, and (iii) pooled accuracy. We define pooled accuracy as the percentage of spans that were mapped to the correct CUI by any of the top 10 systems. In our analysis, we first preprocessed the text spans to remove possessive pronouns tagged alongside the concept (e.g., 'her diabetes' → 'diabetes'). The training dataset consists of 6684 annotations and 2331 unique CUIs, but the 100 most common CUIs constitute 31.5% of the annotations. Since the data set is heavily right-tailed, we evaluate performance for rarer terms by focusing on the following subsets of data:

- All spans *(All)*
- Multi-word spans *(Multi-word)*
- Spans where the corresponding text is unseen in the training set *(Unseen Text)*
- Spans where the mapped CUI is unseen in the training set *(Unseen CUI)*
- Spans where the text isn't a preferred name or synonym in UMLS *(Not Direct Match)*
- Spans mapped to the 100 most common CUIs *(Top 100 CUI)*
- Spans where the text is more commonly mapped to a different CUI *(Unpopular CUI)*

Unpopular CUI measures how well systems do on rarer CUIs, when there are other plausible options. For instance, in the training set, `pt` appears 17 times: 15 times as 'Prothrombin time (C0033707)', once as 'Physical therapy (C0949766)', and once as 'Posterior tibial pulse (C1720310).' A system that always defaults to 'Prothrombin Time' would have 88% accuracy but no disambiguation power.

Table 2.1: Performance of the top 10 systems on the entity normalization task on the MCN corpus. Accuracy is measured on different subsets of the test data and by the 3 metrics described in Evaluation Methodology

Test Subset	**# Examples**	**Max Acc**	**Average Acc**	**Pooled Acc**
All	6925	85.2%	81.1%	92.7%
Top 100 CUI	1891	95.3%	91.1%	98.1%
Multi-word	2755	81.9%	74.3%	91.3%
Unseen text	2910	69.5%	60.7%	86.0%
Unseen CUI	2067	69.3%	61.3%	81.9%
Not direct match	1682	64.4%	50.9%	84.0%
Unpopular CUI	114	7.0%	2.3%	14.0%

Additionally, we investigate how accurately different concept categories, as defined by the UMLS Semantic Types, are normalized. We only consider the categories with at least 50 instances in the test set.

Separately, we analyze the performance of cTAKES and MetaMap. These systems were not trained for this specific task or dataset but are built to provide end-to-end annotations, not just entity normalization. Therefore, to evaluate leniently, we deemed a system correct if it output any span with the correct CUI that overlapped with the test span. If the true span did not have a CUI, any output without the exact span was treated as correct. We didn't factor in any additional spans it might have tagged.

2.3.3 Analysis Results

While overall normalization accuracy is relatively high, it is much more brittle if we look at performance on subsets of the dataset, as detailed in Table 2.1. In particular, if a text span is more commonly mapped to a different CUI, it is very rare that systems recovered the true underlying CUI. Examples everyone missed were for 'Cold Sensation (C0234192)' and for the physiological process of 'Diuresis (C0445398)' where all participants chose the term for 'induced diuresis.' Some fraction of errors came from choosing a concept in the wrong type hierarchy, and many are essentially pedantic, e.g., everyone mapped `tube feeds` to 'tube feeding diet (C0311131)' when the gold label was 'tube feeding of patient (C0041281).'

We find that performance is also variable across the UMLS Semantic Types. For some types, the maximum accuracy was nearly perfect: Pharmacologic Substance (93.9%), Sign or Symptom (94.3%), Quantitative Concept (95.8%). Maximum accuracy was significantly worse for Therapeutic of Preventive Procedures (77.5%), Medical Devices (78.7%), and Body Substances (73.3%). A full table of these results is provided in Appendix B.

MetaMap and cTAKES underperform compared to competition systems. Over the entire test, accuracy was 59.0% for MetaMap, 47.1% for cTAKES, and 64.6% for both pooled together. This is consistent with past findings in which MetaMap yielded poor performance on the SemEval task for disorder/disease recognition [221]. The poor performance of cTAKES can be partially explained by the fact that its Default Clinical Pipeline is not trained to tag adjectives (e.g., severe, mild). However, both systems are primarily disadvantaged by the fact that they are end-to-end systems. For instance, competition participants were provided with the span `breast on the left side` which mapped to the CUI for 'left breast'. In contrast, the end-to-end systems needed to first identify the span themselves, before normalizing them. They only identified the span `breast`, and not the full span, and therefore tagged with the less specific CUI for 'breast.'

2.4 Annotation Framework

Our analysis clearly found that the performance of clinical normalization systems leaves much room for improvement and that properties of existing corpora make it difficult to evaluate end-to-end normalization. In real pipelines, recognition and normalization are usually not decoupled, and as a result, it's important to be able to evaluate fairly in that setting. Therefore, our goal here was to develop an annotation framework to overcome the limitations of past corpora to allow for a more robust, flexible evaluation.

2.4.1 Framework Development

The annotation framework was primarily developed by medical residents on our team. We iterated from existing annotation guidelines by annotating notes (from the i2b2/n2c2 dataset and the MIMIC-III Critical Care database from Johnson et al. [120]), addressing inconsistencies and ambiguities that arose, and noting unannotated information of particular clinical relevance for downstream systems. Our final framework was grounded both by past issues noted by Luo et al. [164] and our own observations.

2.4.2 Amended Framework

We summarize the three largest changes made to existing annotation paradigms changes in Table 2.2. Additionally, we further explain the motivations behind each of the three significant framework shifts. We attach the full annotation framework in Appendix C.

Table 2.2: The three significant changes in our annotation schema, in response to the main limitations we found in current annotation frameworks.

Rule	Justification	Examples
Tag multiple CUIs for a single span if they seem equally valid	We no longer need rules on which CUI to pick. Such rules can be arbitrary and are hard to make universal.	Sputum → Sputum specimen (C0444159) Sputum (C0038056) Stent→ Stent, device (C0038257) Vascular stent (C0183521)
Tag subconcepts if they have a medical meaning	Concept boundaries are subjective and UMLS is inconsistent, but tagging subconcepts removes ambiguity and reintroduces consistency (e.g., for Figure 3)	Severe Asthma Exacerbation: Severe asthma exacerbation → C00038218 Severe → C0205082 Asthma → C0004096 Asthma exacerbation → C0349790
Tag all medical concepts (e.g., anatomical terms, normal findings)	Problems, tests, and treatments are hard to delineate and are insufficient to understand a patient's trajectory.	NSR → Normal Sinus Rhythm (C0232202) Rheum → Rheumatologist (C033489) PEERL → Pupils Equal and Reacting to Light (C1261138)

Change 1: Tagging Multiple CUIS due to Term Redundancy

The UMLS vocabularies contain redundant terms. For example, there exist separate terms for *Post extubation acute respiratory failure requiring reintubation* and *Acute respiratory failure requiring reintubation*. However, if a patient requires reintubation, they are by definition extubated. The MCN corpus approached this problem by pre-selecting concepts in certain hierarchies to have priority over others. For example, many measurements of substances exist in both 'Finding' and 'Observable Entity' forms, and the creators of MCN informed annotators to choose the former, if presented with that choice. However, this could have suboptimal effects on model learning, where, for example, representations for the 'Observable Entity' type are improperly learned. Additionally, such rules are hard to codify over the full domain, so in other cases, an adjudicator over the MCN corpus had to choose one CUI, when the two annotators disagreed. In examining errors from the 2019 n2c2 Shared Task, we did find that many “errors" arose from models essentially choosing synonyms. Therefore, our framework allows annotators to choose more than one concept, where they find relevant. This also allows for a more accurate evaluation, in addition to cleaner signal during model training.

Change 2: Tagging Subspans due to CUI Inconsistencies

Inconsistencies in available UMLS terms complicate generate universal rules about concept boundaries. In the example in Figure 2-2, we show the concept `Severe asthma exacerbation` has a term in SNOMED whereas `severe COPD exacerbation` does not. Additionally existing public clinical entity corpora only tag the CUI corresponding to the longest span of text. Therefore, in the prior example, in one case, `Severe asthma exacerbation` would be tagged, and in the other only `COPD exacerbation` would be tagged. Then, if our model trained on `COPD exacerbation`, our model would likely only tag `asthma exacerbation`. Under current paradigms, this would count as an incorrect tag. As a result, in our framework, we additionally tag subspans corresponding to valid medical subconcepts, as shown in Table 2.2. This change would increase the artificially

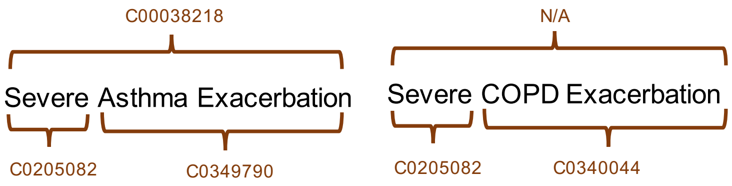


Figure 2-2: Examples of compound concepts represented inconsistently in SNOMED. While there is a term for 'Severe Asthma Exacerbation', there is no term for 'Severe COPD Exacerbation.' Existing datasets only tag the longest concept available in the vocabulary, but this makes it difficult to annotate concepts consistently.

deflated performance numbers from end-to-end systems in our prior analysis, where e.g., they were counted as incorrect for normalizing `breast` and not `breast on the left side`. Further, prior work has shown that using subspans produced by cTAKES (as opposed to only the longest spans) leads to superior predictive performance [88].

Change 3: Increasing Clinical Concept Coverage

Finally, we found that problems, tests, and treatments are insufficient to describe clinically relevant information. Problems only cover abnormal findings, but normal findings are necessary to place abnormal findings in context. Further, the definition of a problem is dependent on the history of a patient and the degree of the finding. For example, the s3 heart sound is normal in children or athletes, but can be pathologic in older patients. We face similar issues in delineating treatments, when we may also want to capture a lack of intervention. Consider a note that said *Hypoxemia:... on 3-4L NC [nasal cannula] throughout the day yesterday, weaned to RA [room air] this AM.* `RA [Room air]` isn't a treatment, but it is clinical terminology that is critical to understanding that the patient's oxygen requirement changed.

Finally, outside of issues delineating the three categories, there is other clinically relevant information essential to a patient's narrative, such as anatomical terms and clinical abbreviations. For example, we may return to the example of `pt` which in the MCN dataset could refer to prothrombin time, physical therapy, or posterior tibial pulse. In addition, `pt` is a common abbreviation for patient, and since it is not a problem, test, or treatment, it would never be seen in training; as a result, a downstream end-to-end system would likely face issues. As a result, we believe it is important for datasets to cast a wider net in terms of the clinical terms they tag.

2.4.3 Comparison to MCN

To provide a point of comparison, we compared a note tagged with our schema versus one tagged with the schema from i2b2/MCN. Under our framework, the annotator identified 170 spans with 123 unique CUIs; in MCN, the note had 72 spans with 63 unique CUIs. The changes in our schema manifested in the following ways:

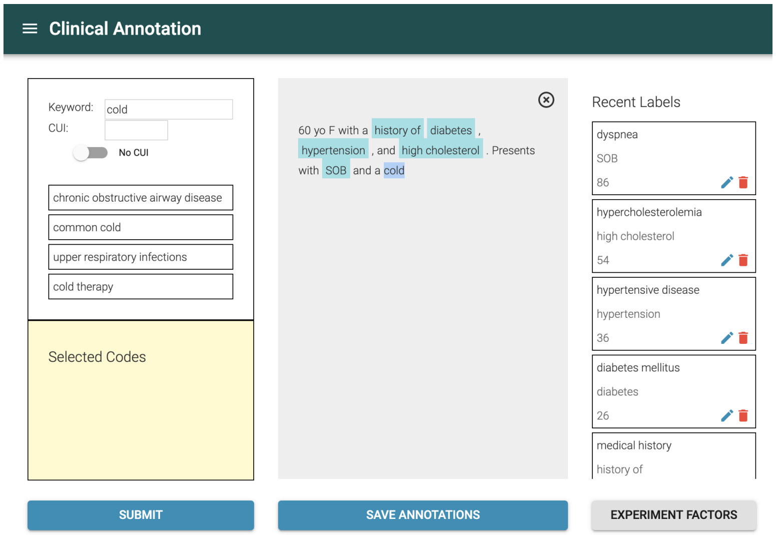
- `Jaundiced` was tagged both as ‘icterus (C0022346)’ and ‘yellow or jaundiced color (C0474426).’
- The note contained `place percutaneous drains to decompress his biliary tree`. The span in the MCN corpus was `percutaneous drains`, which was split into ‘percutaneous (qualifier)’ and ‘drain (device).’ In addition to those, we tagged `place percutaneous drains` as ‘percutaneous transhepatic insertion of biliary drain (procedure).’
- Additional annotations included sections of the note (e.g., `Review of Systems`), normal findings (e.g., `stable condition`), and medical descriptions (e.g., `jaundiced`).

2.5 Validation of Framework

Next, we “validate" our framework to show it can yield consistent annotations and that it had the intended effects on evaluation when applied to real world systems.

2.5.1 Annotating under our Framework

Two annotators tagged portions of notes, using the MIMIC-III database, so the resulting dataset could be open-sourced [120]. They tagged both the History of Present Illness and Assessment and Plan sections of both physician notes and discharge



Keyword: cold

CUI:

No CUI

chronic obstructive airway disease

common cold

upper respiratory infections

cold therapy

Selected Codes

60 yo F with a history of diabetes, hypertension, and high cholesterol. Presents with SOB and a cold

Recent Labels

dyspnea	SOB	86
hypercholesterolemia	high cholesterol	54
hypertensive disease	hypertension	36
diabetes mellitus	diabetes	26
medical history	history of	

Figure 2-3: Custom in-house annotation tool, an early prototype of PRAnCER described in Chapter 3. Annotators highlight the span they want to tag in the middle panel (pictured: cold), and the panel on the left automatically provides suggestions for CUIs, which can be scrolled through and selected. If the suggestions provided are insufficient, an annotator can use the UMLS browser and directly enter CUIs instead. The right panel shows the labels that were already tagged and at which character locations: the pencil allows for edits and the trash can for deletions.

summaries, since these include the bulk of the narrative and the least auto-generated text. 11 note sections were tagged in all. Annotation was conducted using an in-house annotation tool, which is shown and described in detail in Figure 2-3.

2.5.2 Agreement Evaluation

We evaluate concordance between the two annotators by measuring span agreement, and for the concordant spans, CUI agreement. A total of 1571 spans were tagged (1288 by Annotator 1, 1435 by Annotator 2). This corresponds to a Jaccard similarity of 0.73 for unadjudicated span selection. Existing clinical entity recognitions datasets have not released agreement numbers for span selection, so we are unable to contextualize this figure; they merely state that they use two annotators, followed by an adjudicator. On concordant spans, annotators matched on at least one CUI 82.6% of the time (sans adjudication). The MCN dataset reported 67.4% CUI concordance between annotators pre-adjudication. This is not an apples-to-apples comparison, because by additionally tagging shorter subspans of compound concepts, our system benefits from a smaller proportion of compound concepts. However, we also consider this a strength, since it allows compound concepts to be represented in multiple ways to account for inconsistencies in UMLS.

2.5.3 Evaluation of Real World End-to-End Systems

Finally, we wanted to use the dataset we created to benchmark MetaMap v4.0.0 (rule-based) and the cTAKES Default Clinical Pipeline (hybrid rule and ML model), since they are two common systems used in the machine learning and healthcare community, and past comparisons have only focused on narrow concept types. We examine recall on spans in our dataset, since tools like MetaMap also tag terms like `Date`. We can see results of this analysis in Table 2.5.3. For the purpose of the analysis, we considered a union of the annotations created by the annotators, ignoring those spans which did not map to a CUI.

The precision/recall tradeoffs between the two systems are apparent. More inter-

		MetaMap			**cTAKES**		
Semantic Type	*# in Data*	*% Spans Correct*	*% CUIs Correct*	*CUI Precision*	*% Spans Correct*	*% CUIs Correct*	*CUI Precision*
All types	1271	64%	48%	75%	52%	46%	89%
Sign or Symptom	168	68%	62%	92%	83%	82%	99%
Disease or Syndrome	164	68%	52%	77%	64%	62%	97%
Finding	130	57%	43%	76%	41%	31%	74%
Pharmacologic Substance	113	80%	75%	94%	86%	84%	98%
Organic Chemical	102	79%	77%	98%	88%	88%	100%
Body Part, Organ, or Organ Component	84	40%	22%	55%	72%	66%	92%
Therapeutic or Preventive Procedure	71	58%	30%	51%	38%	24%	63%
Pathologic Function	53	62%	49%	79%	83%	68%	82%
Diagnostic Procedure	43	70%	47%	67%	51%	44%	86%
Temporal Concept	41	54%	46%	86%	0%	0%	N/A

Table 2.3: Performance of MetaMap and cTAKES on the whole dataset created under our framework, and subdivided by the ten most common Semantic Types. *% Spans Correct* indicates what percent of spans were correctly recognized by the model, *% CUIs correct* indicate what percent of spans were correctly recognized with a correct CUI, and *CUI precision* indicates the percent of correctly identified spans that were normalized correctly.

estingly, we find huge discrepancies of accuracies among Semantic Types between the systems. This highlights the utility of benchmarking across a diversity of terms that reflect downstream usage.

Finally, we also examine the 174 spans that contained at least one subspan and were not contained within any larger span. These are exactly the compound concepts that are inconsistently included in medical vocabularies. Of these, cTAKES only recovered the span in 74 cases. However, of the 100 cases with a missed span, in 54 of them, cTAKES did correctly match a CUI for one of the subspans. In existing schemas where only the longest span is tagged, these 54 (31%) cases would have all been counted as incorrect, which vastly misrepresents the true performance of cTAKES. We don't see as drastic a bump with MetaMap, since perhaps in contrast to cTAKES, it is entirely rule-based.

2.6 Discussion

We have shown that performance of entity normalization is far from robust, and accuracy numbers are hugely bolstered by the existence of a few common, easier-to-map terms. As we have shown, concentrating on certain subsets of the data can lead to vastly different metrics. As a result, performance would be highly variable for downstream applications depending on the entity of interest. This holds true both for the normalization systems from the n2c2 Shared Task and for the widely used MetaMap/cTAKES. Being able to compare systems with a reference dataset is crucial for deciding how to set up a clinical machine learning pipeline. Our annotation framework could be used to create that dataset, since it allows for more nuanced evaluation.

In examining the discrepancies between our annotators, we notice they often arose because of minute differences between concepts, e.g., 'malabsorption' versus 'malabsorption syndrome.' In these cases, note context is insufficient to pick up on subtle differences in definition, which are also generally of little clinical consequence. A system that retrieves either of these should be assessed as correct.

Perhaps most importantly our schema allows for a transition towards end-to-end learning and evaluation. Let us consider an example created from our framework where an annotator tagged `dirty UA` as 'urine screening abnormal (Finding)' and `UA` as 'urinalysis (Procedure).' We can give partial credit to a system that only tags `UA`. In contrast, under current schemas which only tag the longest concept span, normalizing only to 'urinalysis' would be considered incorrect. Not only does this discount the model's true performance, but it also adversely affects training. Further, in this case, the subspan `UA` is of a different Semantic Type (Procedure) than the whole span (Finding). If a researcher only wanted to extract procedures, under our framework, they won't lose out on those spans where procedures are a subspan of a larger concept. Therefore, our schema allows for flexible training of a subset of concepts of interest.

Limitations

Given that we tag spans in addition to subspans, our annotation framework does greatly increase the number of terms that need to be tagged per document. However, this extra effort is partially offset by the decreased cognitive load of deciding where a span boundary is. Further, by providing suggestions upon highlighting a span, the clinical annotation tool already greatly speeds up the time needed to annotate. Moving forward, we are currently experimenting with having the tool auto-tag unambiguous cases, which the user could simply choose to accept or reject. Since our analysis of n2c2 shows that we can already extract certain terms (e.g., `hypertension`) and Semantic Types with high confidence, this allows the annotator to focus only on the most difficult 20-30% of cases. In manual examination of spans tagged only by a single annotator, differences often arose due to annotation fatigue, rather than misunderstanding of the framework. For example, an annotator would not tag a text span they had previously tagged before in the note. This fatigue could be alleviated by the addition of automatic suggestions.

However, with suggestions and auto-tagging, it is important to understand the sources of bias we could introduce to our resulting dataset. An ongoing area of research is assessing the extent to which the presence, ordering, and presentation of suggestions and auto-tags may bias outcomes. Additionally, we hope to evaluate user interface (UI) techniques to mitigate this bias. There is a need for larger annotated clinical datasets, and smart computational tooling can decrease the time, effort, and cost needed to create them. To that end, we open-sourced our clinical annotation tool, as well as the annotations generated by this chapter. See `https://github.com/clinicalml/mimic_annotations`.

We do acknowledge that the annotation guidelines were developed and evaluated using notes written in a critical care context, so that we could open-source the resulting MIMIC data set. Future work remains to evaluate the schema in the context of other clinical settings and note types.

More broadly, approximately concurrently with the writing of this chapter, several clinical entity recognition systems were released by Amazon, Google, and Microsoft [1]. As one data point, Guzman et al. [99] assessed the capability of Amazon Comprehend Medical at medication attribute extraction and found it underperformed systems from the i2b2 and n2c2 challenges. However, further work needs to be conducted to continuously benchmark these newer systems as well.

Additionally, clinical concept recognition is only the first step towards parsing the information in clinical notes. The community has long recognized that successful reconstruction of a full patient narrative requires additional tasks, e.g., relation extraction and temporal reasoning, and has constructed challenges around them [263, 277]. Conducting a similar detailed breakdown on state-of-the-art systems for those tasks is crucial to help us understand failure modes before deployment.

2.7 Conclusion

Medical notes offer us an incredibly rich view into patient narratives, but they have remained a largely untapped resource for clinical and machine learning research. Clinical studies often resort to manual chart review, which is time-consuming and difficult to scale. While automated concept extraction systems provide a way to exploit clinical text, as we have demonstrated in this chapter, they cannot yet be reliably deployed.

Developing truly robust and accurate clinical entity recognition and normalization algorithms will require both substantially more labeled training data and automated evaluation metrics that account for the subtlety and ambiguity of the task. We present a new annotation framework and a small-scale annotated dataset for evaluation of end-to-end concept recognition and normalization. The time is ripe for the field to invest in substantially larger labeled training sets to spur new machine learning approaches for clinical entity recognition and normalization. Future work should consider incorporating the new annotation framework as the basis for such a data set.

---

[1] `https://aws.amazon.com/comprehend/medical/`, `https://cloud.google.com/healthcare-api/docs/concepts/nlp`, `https://learn.microsoft.com/en-us/azure/cognitive-services/language-service/text-analytics-for-health/overview?tabs=ner`

Chapter 3

Decision Support Tool for Clinical Entity Annotation

Acknowledgement of Co-authors I would like to acknowledge my undergraduate mentee Ariel Levy, who provided the software development for the bulk of the annotation platform described in this chapter.

3.1 Introduction

In the previous chapter, I lay out the middling algorithmic performance of existing clinical concept extraction systems and the need for a greater amount of labeled data, both for training and evaluation. The relative lack of existing labeled data can be attributed to the domain expertise required and the tediousness of the task, which is exacerbated by the lack of tooling made for clinical concept annotation. Clinical vocabularies can consist of hundreds of thousands to millions of concepts, and manually searching for each concept identifier can lead to prohibitive slowdowns.

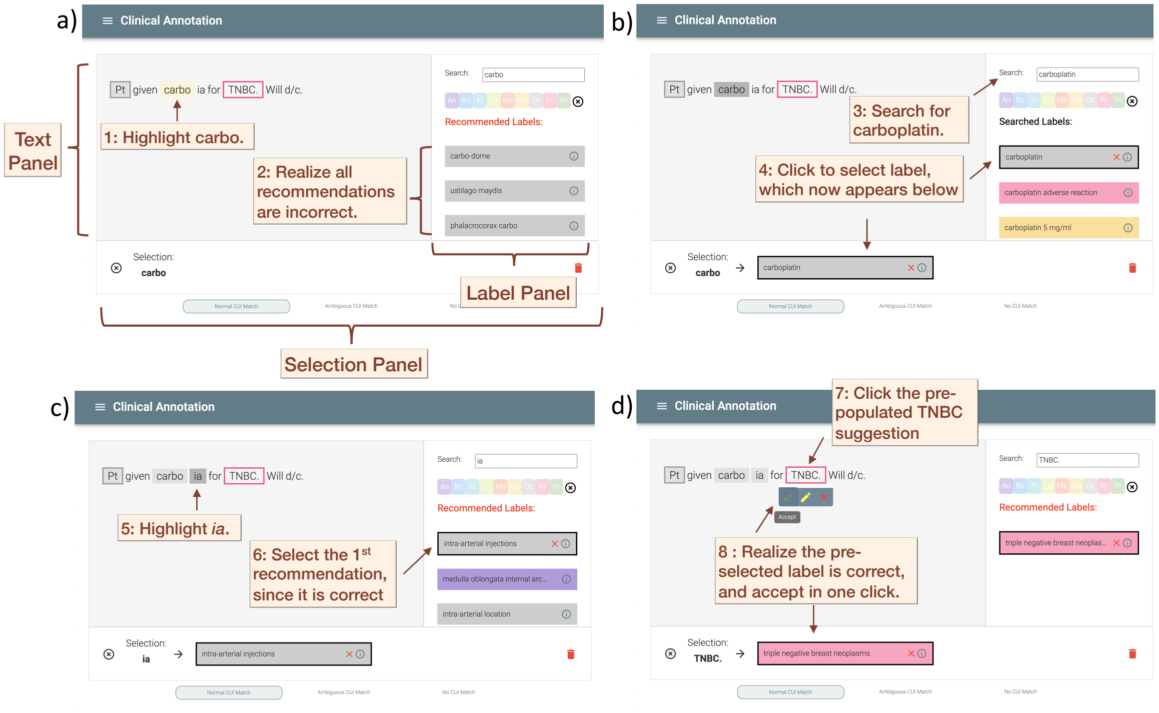
To address this gap, in this chapter, we introduce an open-sourced annotation platform PRAnCER, built specifically for concept annotation. In addition to being able to handle matching to large ontologies, PRAnCER provides two major forms of automated decision support. The first are label recommendations, which map a span that a user selects to a list of ten model-proposed concept labels. The user can choose to go with one of the recommendations or decide to search further. The second form is fully pre-populated annotation suggestions, which suggest both the text span itself and the corresponding label. These are appropriate for high-precision cases, in which a text span can be easily mapped to a single label (e.g., ‘hypertension’). In contrast, MS might refer to the title ‘Ms.’, mitral stenosis, or multiple sclerosis. Given the uncertainty, we would not want to provide any pre-population.

3.2 Related Work

There are many ways to annotate, explore, and understand text corpora, but one common preprocessing step is to identify the entities within a document; in the general domain, these tend to be names, places, and organizations [33]. For example, Jigsaw is a system for text exploration (e.g., articles, reviews); as a starting place, it algorithmically pre-tags entities in documents, and then allows users to make corrections to these annotations [91]. Historically, identifying clinical entities is particularly tricky, since the language in doctors’ notes can essentially be considered its own dialect, and there are many overloaded terms, e.g., `MS`. Deciding what counts as a clinical entity (finding, disease, procedure, treatment, lab test) and what defines an entity’s boundary is much more ambiguous than in the general domain; this is because concepts are not just proper nouns and often overlap with one another.

Due to the importance of understanding concepts and entities in text, there are a host of tools focused solely on entity annotation; however, these tools were created for tasks with a more limited number of labels (on the order of <30-40) that can all be simultaneously displayed. Many of these tools (WordFreak, GATE, BRAT, WebAnno, Knowtator and YEDDA) do include system suggestions and pre-annotations [178, 312, 260, 126, 315, 197].

There have been several open-sourced datasets consisting of our same clinical text annotation task, or slight variants on it [4, 66, 164, 264, 212]. These datasets have been created via annotation tools such as MAE without decision support for label selection; annotators have to resort to external websites to search the label space



1: Highlight carbo.

2: Realize all recommendations are incorrect.

3: Search for carboplatin.

4: Click to select label, which now appears below

5: Highlight *ia*.

6: Select the 1st recommendation, since it is correct

7: Click the pre-populated TNBC suggestion

8 : Realize the pre-selected label is correct, and accept in one click.

Figure 3-1: Our clinical annotation system in the midst of annotating our sample sentence. Panel a) shows the text, label, and selection panels. In the above, *Pt* has already been annotated; the user annotates *carbo* using the search feature in panels a) and b), *ia* using the automatically surfaced label recommendations in panel c), and *TNBC* using a pre-populated annotation suggestion in panel d).

[261, 194]. As a result of the tediousness of the task and lack of specialized tooling, the datasets are small, with the largest being on the order of hundreds of notes.

3.3 System

Due to the complexities and idiosyncrasies of the clinical annotation task, we built a custom annotation platform[1]. The platform is based on top of React, Typescript, Node.js, and Flask. In this section, we first present a walkthrough of representative usage of the platform, and then elaborate on the system behind automated decision support.

---

[1] An up-to-date version of our platform can be found at clinicalml.github.io/prancer.

Usage Walkthrough The platform consists of three main panels, shown in the top left of Figure 3-1: a text panel, a label panel, and a selection panel. Users begin by looking at the text displayed in the text panel; in Figure 3-1a), we see that the shaded box around `Pt` indicates it has already been annotated, and the user highlights the next clinical span, `carbo`. Upon highlighting, the selection panel shows that `carbo` has been chosen, and the label panel shows a list of automatically generated recommendations. The user notices that none of the provided options are correct, then in Figure 3-1b), uses the search bar to explicitly search for *carboplatin* instead. A list of search results appear, color-coded by concept type (e.g., pink for *Problems*, grey for *Other*); we use the 11 concept types described in Patel et al. [203]. Our user would notice that the first suggestion is correct and click on it, placing the label in the selection panel. The user could continue searching for additional labels, or move on. In Figure 3-1c), they highlight *ia* and the first displayed label is correct, so they can select it and move forward. In Figure 3-1d), the user then comes to `TNBC`, which already has a pink outlined box around it, indicating that the system has pre-populated an annotation for `TNBC`. Upon clicking the box, the system displays what it has identified as the likely correct label, which it auto-populates in the selection panel. The user then can choose to accept this span and label combination in a single click on the green checkbox that appears as a dropdown; alternately, the user could choose to modify the label via the yellow pencil or to discard the result via the red 'X'. At any time, the user can return to edit any annotation by clicking on its surrounding box.

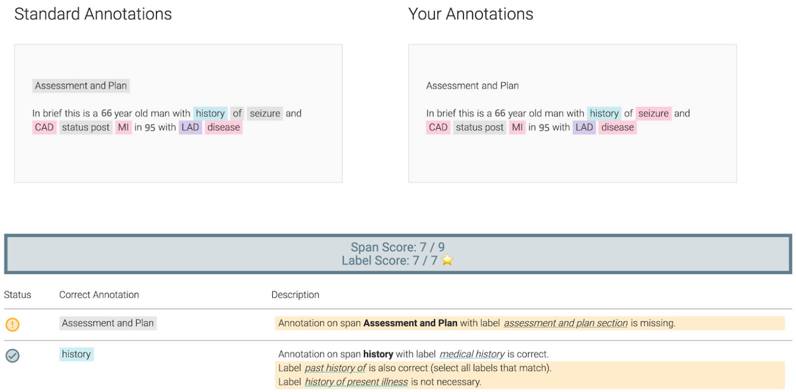
We adopted the multi-panel view to mimic other interactive writing/tagging applications (e.g., Grammarly) where text spans are highlighted in situ to indicate the presence of recommendations, which are then shown in a side panel [117]. In earlier prototypes, we explored alternate design decisions. For example, we considered displaying recommendations in-line via a drop-down menu and displaying existing labels in-line; we found that the former idea obscured surrounding text that was useful for ascertaining context, and the latter cluttered the screen with minimal benefit. We also explored alternate recommendation confidence indicators and suggestion utilities such as *Recently Used*, and our final interface was a synthesis of these ideas.

Additional Feature Details In addition to the features explicitly walked through above, there are a few more to accommodate the challenges of clinical text annotation. Users can indicate in the selection panel if there is no matching concept, or only an ambiguous match. Further, users do have the ability to select overlapping spans, and when there are multiple annotations for a single word, users may toggle between these annotations. If a user needs more descriptive information on a concept, they can click the *(i)* button on any label in the label panel to surface the official medical definition. Finally, a user can click on one of the colored boxes under the search bar to filter recommendations or search results to one of the 11 concept types; these colors correspond to those used in the search and in the annotation bounding boxes.

In the platform, any addition, deletion, or modification of manual or suggested annotations is automatically saved in a JSON-serialized dataset. Saved features include the character span numbers, the labels selected, the timestamp, and whether the annotation was manual or suggested. For the purposes of our study, we also log all interactions the user has with the platform.

Automated Decision Aid As mentioned in the walkthrough, we introduce two major forms of automated decision aids in our platform. The first is via automated label recommendations given a user-highlighted span (see *ia* in Figure 3-1c); the second is via fully pre-populated annotation suggestions (see *TNBC* in Figure 3-1d). We describe each modality and its motivation in greater depth below.

Automatic label recommendation could greatly decrease the amount of searching users have to do. While there is a large label space (>400,000 concepts), there is also a set of lower-hanging fruit. These are text spans that, once highlighted, are either straightforward to label or narrow down to a set of labels. On their clinical text corpus, Luo et al. showed that when provided with a correct text span, simple heuristic methods can achieve 77% label accuracy; the highest accuracy yet achieved on their data set is 85% [164]. Therefore, while existing algorithms are not sufficiently robust to conduct automatic extraction in clinical workflows, given an identified span, they are sufficiently advanced to narrow down to a set of useful recommendations in the vast



Standard Annotations

Assessment and Plan

In brief this is a 66 year old man with history of seizure and CAD status post MI in 95 with LAD disease

Your Annotations

Assessment and Plan

In brief this is a 66 year old man with history of seizure and CAD status post MI in 95 with LAD disease

Span Score: 7 / 9
Label Score: 7 / 7 ⭐

Status	Correct Annotation	Description
ⓘ	Assessment and Plan	Annotation on span **Assessment and Plan** with label *assessment and plan section* is missing.
✓	history	Annotation on span **history** with label *medical history* is correct. Label *past history of* is also correct (select all labels that match). Label *history of present illness* is not necessary.

Figure 3-2: Tutorial mode built into the platform to train users with rounds of iterative feedback. Gold standard and chosen annotations are displayed side-by-side, with feedback on all annotations provided below.

majority of cases, particularly for frequently occurring concepts. As a result, our system surfaces recommendations automatically on the right label panel when a user highlights a span, as seen in Figure 3-1(a, c). On the backend, the recommendation system is built within a Python wrapper, allowing for easy extensibility to any modern machine learning model as algorithms improve. In our studies, we use the recommendation algorithm detailed in Agrawal et al., which draws on classical information retrieval techniques from NLP [4].

Our platform can also incorporate fully pre-populated annotations, where both the span and the label are provided. As described previously, the field of clinical NLP has already created several systems that attempt to automatically extract concepts end-to-end: both identification of spans and mapping those spans to labels. While these are obviously imperfect, thus necessitating the need for clinical text annotation, they can be used to recover at least half of concepts. Given that a note can easily have hundreds of clinical concepts, these systems could relieve the burden on the human user by potentially allowing them to focus their attention on more difficult cases. In our platform, pre-populated annotations appear as outlined boxes around the suggested span, as seen with `TNBC` in Figure 3-1. On the backend, the pre-populated annotations are computed and rendered before the user begins their annotations. Therefore, latency is not a concern, allowing any concept extraction system to be used to provide the pre-populated annotations. Possible existing concept extraction systems include cTAKES, MetaMap, MedLEE, and scispaCy [15, 236, 79, 188].

Finally, due to the repetition of terms within a note, we additionally implemented a feature that propagated concept annotations to repeat occurrences of the same entity. For example, if a user marks `carbo` at the beginning of the note as *carboplatin*, all future occurrences of `carbo` in that note will appear with a pre-populated annotation for *carboplatin*, that users can again choose to accept, modify, or delete.

Tutorial Finally, we included a tutorial mode in our system, shown in Figure 3-2. As input, the tutorial mode takes in a series of snippets and their gold standard annotations. To begin, users annotate the first snippet using the standard interface shown in Figure 3-1. After, they are brought to the tutorial screen shown in Figure 3-2. It presents the gold standard annotations on the left, the user's annotations on the right, and below, it provides a score based on the number of spans correctly recovered by the user, as well as the number of correct labels. Underneath, it iterates through all the annotations, providing a description of the differences between the gold standard and the user output, if any. Then, the user iterates through steps of annotation and tutorial feedback until the mode is over. The tutorial serves two purposes: (i) to familiarize users with the annotation task and the platform and (ii) to allow them to form mental models of the accuracy of the automated features.

Flexible Usage While PRAnCER is equipped with features specific to the annotation of UMLS concepts (e.g., the ability to query for additional meta-information about any CUI), the ontology files can be swapped out to handle any ontology. For example, I adapted PRAnCER for the work described in Singh et al. [252]. In that work, PRAnCER was used to annotate 21 different measurements from 370 cardiac magnetic resonance imaging reports. Clinician labeling time was estimated at 15 hours, which comes out to approximately 2.5 minutes per report. Using 270 reports as training data and 100 for evaluation, the authors were able to achieve 0.957 AUC at automated measurement extraction. Similarly, PRAnCER is being used in a diverse set of scenarios in ongoing work by others in the informatics community. It remains future work to allow for further flexibility by adding functionality for annotation of relations.

Chapter 4

Assessing the Impact of Decision Support

Acknowledgement of Co-authors I would like to acknowledge my undergraduate mentee and co-first author Ariel Levy, who conducted the user studies and aided in the design and analysis of this study.

4.1 Inroduction

In the previous chapter, I introduced an annotation platform that provides two major forms of automated decision support: label recommendations and fully pre-populated annotation suggestions. This introduction of automation can decrease the cognitive load on human decision-makers, enabling them to focus their attention where it is most needed. This is a wider phenomena, as decision tools have emerged across a variety of disciplines including medicine [284], sports [198], and criminal justice [132]. Since automated systems and humans often have complementary strengths, joint systems can outperform either alone [284].

However, while such hybrid intelligence systems are very promising, it is important to understand the dynamic between the human decision maker and the automated system, a topic of longstanding study [147]. For optimal results, the human needs to understand when to listen to the computer and when to exercise their own agency [21].

If not, the human may develop a misplaced trust in the automation, which could have the adverse effect of degrading their output, as has been demonstrated in prior work [265]. Further, this trust may cause them to lose critical engagement with the task, so they do not have the attention to intervene when necessary, as has been shown in self-driving car examples [108]. However, this phenomenon has been less studied for domain experts, who may be more confident in their own abilities and more skeptical of automation [154].

In this work, we use PRAnCER to study whether for clinical experts, their expertise mediates their interactions with automation. In particular, we investigate whether domain experts display the misplaced trust and loss of agency that has been described in other past work, and see whether they remain critically engaged in the task at hand when using automation. Recommendations could keep users from searching further when needed and decrease their accuracy, and the pre-filled suggestions allow us to study whether they lose any autonomy in creating new annotations.

We run a two-stage user study on 18 clinicians from 9 different United States medical institutions, in which we artificially vary the extent and accuracy of both of the aforementioned decision aids. In contrast to previous work with discrete, often synthetic tasks, our label space includes 400k+ concepts, allowing us to simulate when models are only slightly off (missing the correct label, but presenting similar adjacent labels), instead of scenarios where the model is entirely incorrect, reflective of real-world error modes. In doing so, we are able to pinpoint specific impacts of automation errors. We investigate how reliant users are on automation by examining how they deal with poor recommendations and the agency they show in annotating additional concepts. We analyze each annotator's behavior and outcomes over approximately eight hours of annotation each, allowing us to account for effects over long time scales, such as tedium. Through this analysis, our paper makes three contributions:

- We find that domain experts (n=18) are sufficiently engaged to notice when system label recommendations are inadequate. Due to the size of the label space, we were able to measure how often they chose to search, an objective proxy for trust in the recommendations. Our users have a strong intuition of when to explore further, but if this intuition is violated and a correct label isn't present when they might expect it to be, they accept substandard label choices for spans they selected.

- When presented with fully pre-populated annotations, we find that our domain experts are more hesitant to exercise agency. While they do change incorrect labels, they are slightly more hesitant to intervene with incorrect spans, and they demonstrate less initiative in the creation of additional spans. Moreover, through exit surveys, we find that they thought they were being thorough and do not note this shift in their own behavior.

- We analyze error patterns of our domain experts and find that trust in automation correlates across suggested labels and spans. However, misplaced trust and loss of agency do not correlate with each clinician's prior demonstrated competency at the task. We also find that human error patterns differ from algorithmic error patterns, indicating the utility of combining the two.

Our results, compiled from over a hundred hours of logged interaction between clinicians and decision support, strongly indicate that domain experts can fall susceptible to risks in human-AI teams. Our detailed characterization of users' behavior can help inform the design of user-facing systems for data collection as well as the machine learning models that ingest that data.

4.2 Related Work

4.2.1 Mental Models and Model Trust

The idea that effective group work requires an accurate mental model of one's teammates—humans or AI—goes back decades [40, 95]. In interviews, clinicians discussed their information desiderata when onboarding new AI decision support: strengths, weaknesses, point-of-views [38]. These needs closely mirrored the information they use to create mental models of their colleagues when seeking second opinions. It has been empirically shown that users with better mental models of their AI teammates are more successful; in particular, knowledge of the error boundary enables the user to know when to trust automation and when to override it [21, 85]. However, in cases where users do not properly learn when to trust automation, they can become overly reliant on systems [147]. On image recognition tasks, Suresh et al. found that people will trust an incorrect machine decision, even if they would have made the correct decision on their own [265]. Similarly, a study of radiologists showed that while decision aid helped lower-performing radiologists, it actually hindered the performance of the best ones [211]. Further work has shown that users' subjective evaluations of their trust in and relationship to decision support systems does not always align with actual outcomes, highlighting the importance of quantitative studies [37].

Due to the importance of calibrating trust in decision aids, several recent studies have analyzed how the presentation of model decisions modulates trust. They've investigated factors including reported model confidence, model explanations, overall model accuracy, and initial model accuracy [323, 316, 201, 195]. In these studies, users choose to accept or reject the model's output on discrete binary tasks and self-report their level of trust in the model.

In this work, we study a more complex task which (i) has a large label space (>400k), requiring reliance on computational aid and allowing us to measure trust by how often users search, and (ii) requires users to select spans, additionally allowing us to measure agency. Further, the time scale of our study is longer than past work, allowing us to account for real-world effects such as tedium that would likely affect reliance and agency. The primary goal of clinical text annotation is to create data to better train machine learning models. However, if we provide an initial model as decision support and users become overly reliant on that model, their output could closely resemble the model's, instead of the underlying truth. This would lead to models being fed back their own outputs as inputs in future training; this could cause a dangerous feedback loop in which our models become even more confident in their own incorrect decisions [243]. Therefore, it is imperative to understand if our users will properly mediate model errors, or simply reflect them back.

4.2.2 User Agency

Some tasks like image classification have discrete inputs with clear objective outputs. However, for other tasks that require initiative or creativity, there is also the worry that humans will cede their agency to automated aid [106]. For example, in the translation task, there are often multiple equally valid outputs. When translators used an interface that displayed machine-suggested recommendations, they noted that they ceded agency and would conform to the machine's recommended phrasing, even when it did not match their usual style [93]. Kulkarni et al. showed in an experiment on sketching, exposure to examples increases conformity of users' drawings [141]. Similarly, using an ideation task, Siangliulue et al. investigated the effect of aid (example ideas) on user creativity; users who were exposed at consistent intervals to example ideas actually ended up generating fewer ideas than users with no aid at all [250].

In our work, clinical text annotation does not just require mapping mentions to concept labels, but it also requires deciding what terms are clinically relevant concepts and need to be labeled. While automated methods can provide users with a subset of these terms that need to be labeled, users could fall susceptible to similar patterns of tunnel vision, where their attention and mindset become fixed on what has been provided. On the other hand, a note contains over a hundred annotations, and a well-designed suggestion system could decrease the burden on users, enabling them to focus their attention where it is needed most, instead of replicating what is known.

4.3 Annotation

There have been multiple studies of the impact of pre-annotation on outcomes, though the focus is often on efficiency gains [315, 81]. Past literature has also examined effects of pre-annotation on performance, both in standard NLP and clinical settings, but these have had mixed conclusions on the bias induced and the resulting time savings [156, 87, 73, 258]. These works were based on small numbers of annotators (generally $n = 2 - 4$) and focused primarily on overall agreement of an annotator with the gold standard and other annotators, rather than directly analyzing when they accepted incorrect aid. Further, our work differs due to (i) the large label space we are mapping to and suggesting over, which intensifies the dynamics around trust and mental models, (ii) the wider task definition (e.g., compared to part-of-speech tagging or identification of a few specific symptoms), and (iii) our purposeful introduction of certain modes of errors.

4.4 Stage 1: Label Recommendations

In this first stage of the user study, we test the effect of presenting automatic label recommendations once a user highlights a span of text. Recommendations could decrease the burden of the user searching over the large label space, especially since algorithms could present a correct label over 80% of the time [164]. However, there is a concern that in the presence of recommendations, users will become overly trusting and accept substandard labels instead of taking the initiative to search further. This could lead to a feedback loop if the data created were used for updating the automation model. Therefore, in this stage, we investigate whether our domain experts form appropriate intuition of when to search further, or whether they become complacent in the presence of recommendations.

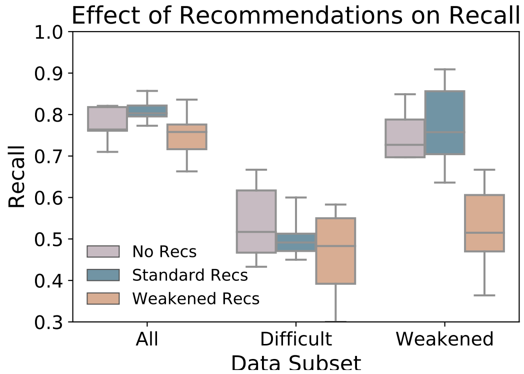
4.4.1 Experimental Design

To ensure a consistent and sufficient clinical background, we required users to have completed at least two years of medical school and to have experience with clinical notes in United States healthcare settings. We recruited 18 clinicians via Twitter and email lists. Our users (8 men, 10 women) consisted of 4 medical school graduates, 6 fourth years, and 8 third years; they came from 9 different medical institutions across the United States. Users were compensated $20 per hour for their time, and each spent between 4 and 5 hours total on this stage. All components of the study were conducted virtually and were ruled IRB exempt.

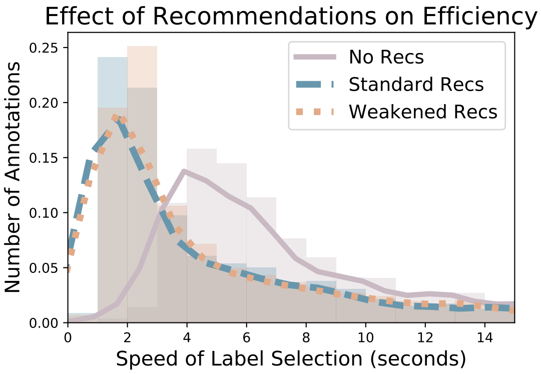
Our study had a multiple factor design, where we evaluated the performance of users (i) across different recommendation modes and (ii) across different annotation task difficulties that naturally arose in the task. First, users were assigned randomly to one of three modes: the *None* mode, with no recommendations (5 users), the *Standard* recommendation mode (6 users), or the *Weakened* recommendation mode (7 users); the number of users per mode differed due to unanticipated changes in clinicians' schedules. To isolate the effect of recommendations, we did not include any pre-populated annotations in this stage. The *Standard* mode presented the model recommendations in all cases. The *Weakened* mode presented the same recommendations as the *Standard* mode, but with the correct label removed in 25% of the nontrivial examples. Examples were considered nontrivial if the text did not directly match a concept or any of its synonyms in the medical vocabulary. In other words, we would only remove the correct recommendation if there would be a better search query to find the concept; the removal of a concept was done consistently for all instances of a term across this stage. In all, a correct label was presented in one of the ten displayed recommendations 83% of the time in the *Standard* mode and 73% in the Weakened mode. Across all modes, we used the search provided by the Unified Medical Language System API [103].

To train users, we individually gave each of them a 30-minute presentation, detailing the annotation rules they should follow, taken from [4]. Users were encouraged to interject and ask for clarifications, as needed. Then, we had them annotate a sequence of eight clinical snippets using the tutorial in Figure 3-2, while we were present. Each user conducted the tutorial in their assigned study mode, so that they could be introduced to the strengths and weaknesses of the automation in their mode. We ensured the tutorial modes had approximately the same proportions of correct and incorrect recommendations as the full notes to allow users to build appropriate mental models.

Users then tagged 3 sections from de-identified clinical notes from the MIMIC-III Critical Care Database [120]. The order of notes was randomized between users. Each user annotated the first note of each stage live over Zoom, talking out loud, and then



(a) Total recall compared to the gold standard. Recall shown over all annotations, “Difficult" annotations (where recommendations do not surface a correct answer), and "Weakened" annotations (where the correct answer was removed from *Weakened* mode).



(b) Histogram and density curve for time taken to label a span under different recommendation modes (median of 6 seconds in *None* and 3 seconds in *Standard* and *Weakened*).

Figure 4-1: Accuracy (total recall) and efficiency (time to label) results for users with label recommendations (*Standard* and *Weakened* modes) and users without (*None* mode).

annotated their next two notes asynchronously, with screen recording. Following each stage, users received a followup survey asking about their workflow, specifically their confidence in and reliance on the automated decision aid features.

For evaluation, we compared to the gold standard released by Agrawal et al., which contained 335 annotations over the sections used in this stage [4]. Since users are allowed to select multiple labels, we treated an annotation as correct if any selected label were among the gold-standard labels. Further, since there may be multiple correct labels and the gold standard may not have contained them all, we manually checked whether any additional user labels were correct, and if so, added them to our gold standard. This process was conducted blind to the user and their mode. Further, we excluded all spans where the label was considered ambiguous in the gold standard.

We analyze several dependent variables: users' accuracy, the speed of clinical annotation, and the actions taken in the platform (e.g., choosing to search). Accuracy was evaluated via *span recall*, the proportion of annotations in the gold standard that users annotated the span for, *total recall*, the proportion of annotations in the gold standard that users got the correct span *and* correct label for, and *label accuracy*, the percentage of time users chose a correct label for a set of spans. In our mixed effects design, our other factor was annotation difficulty. In evaluating user accuracy, we consider multiple subsets of annotations, including "Easy" examples (the examples in which the correct label is provided in both recommendation modes), "Difficult" examples (the examples in which the correct label is not provided in the *Standard* mode), and "Weakened" examples (the examples in which a correct label is provided in the *Standard* mode but not in *Weakened*.) We also examine performance across clinical concept types.

4.4.2 Results

For our multi-factor design (user mode and annotation difficulty), we first conduct an Aligned Rank Transform (ART) procedure [303]. We find that total recall is significantly affected by mode ($p<0.03$), annotation difficulty ($p<1e\text{-}16$), and their interactions ($p<1e\text{-}4$). Both factors and their interaction additionally significantly affect how often users search on the platform ($p<1e-8$ across all three). We now deep-dive into pairwise comparisons and implications behind results.

Recommendations increase annotation efficiency and seem to decrease tagging fatigue. Across all annotations, users in both recommendation modes are able to find labels far quicker (median of 3 seconds) than users without recommendations (median of 6 seconds) as displayed in Figure 4-1(b); the median time for both recommendation modes is statistically significantly faster than users without ($p<0.05$ across both, adjusted two-sided Mann-Whitney U test). When a correct answer is provided in the ten displayed recommendations, users only require a median of 2 seconds. On the set of "difficult" examples where users are provided only with incorrect recommendations, they take a median of 10 seconds, the same as those without recommendations take on the "difficult" set. Further, the users in *Standard* mode create an average of 12% more annotations than those in *None* mode (375 vs 337), a statistically significant increase ($p<0.02$, two-sided Mann-Whitney U test). Users in *Weakened* mode were in the middle of both, with an average of 353 (not significantly different). We hypothesize that the decreased workload stemming from recommendations led to a lower cognitive load for users, decreasing their tagging fatigue and enabling them to create more annotations.

Recommendations generally improve recall, and domain experts step in appropriately in spaces where algorithms fail.

As seen in Figure 4-1(a), users in the *Standard* mode had higher total recall over users in the *None* and *Weakened* modes (80% vs. 76% and 76% at median, respectively). The superiority of the *Standard* mode over the *None* mode is not statistically significant, but results indicate that the presence of recommendations does not decrease recall, a prior worry. On the set of difficult examples (where the recommendations do not contain a correct label), users across all modes have approximately the same total recall (a median of 52% for *None*, 49% for *Standard*, and 48% for *Weakened*). One user noted in their survey that they “really appreciated the suggested labels, but ...

these can institute bias due to availability”, the only user to mention such concerns. This user also had the highest label accuracy on the set of “Difficult" terms which may hint that active awareness of bias can help combat it.

We observe that human recall is much more consistent across different types of clinical concepts; this stands in stark contrast to algorithmic methods. For example, algorithmic accuracy of one existing extraction system is around 48% overall, but only 24% for procedures [4]. Meanwhile, humans achieve about 70% accuracy on procedures, compared to approximately 80% overall. This indicates that human errors don’t follow the same error patterns as algorithms, and therefore are adding valuable signal.

Users develop intuition of when the recommendations should surface a correct answer, but label accuracy suffers when that intuition is disrupted.

Here, we investigate whether users recognize when a correct answer is present in the provided recommendations, and when they need to search further. We break down percentages by mode and example type in Table 4.1. On “easy" examples (those where a correct answer was in the recommendations), users in both the *Standard* mode and the *Weakened* mode only searched further 17% of the time and ultimately chose a recommended label 96% of the time. However, on “difficult" examples, *Standard* users searched further 85% of the time and *Weakened* users 82% of the time. Therefore, we see that their search patterns are closely aligned, and they generally learn to search further when necessary. This indicates that users had a strong sense of cases in which the algorithm isn’t surfacing the correct answer.

However, we observe that performance breaks down if there is a "violation of intuition", namely users expect a correct label to appear and it isn’t present due to synthetic removal; again we only removed answers for a random subset of “nontrivial" examples, where there was no direct match, and a better term could be found by searching. As evidence, we examine the performance of users in the *Weakened* mode on the “weakened" examples, nontrivial examples in which the correct labels were randomly excluded from the recommendations. On these “weakened" examples, a set

Mode / Data Subset	*“Easy" Examples*	*“Difficult" Examples*	*“Weakened" Examples*
Standard	17%	85%	15%
Weakened	17%	82%	73%

Table 4.1: The percentage of times users chose to initiate a search. Across modes, searches are rarely initiated when the true label is provided in the recommendations (“Easy"), and are often initiated when the true label isn’t provided (“Difficult"). When a user expects a label to be provided but it is not (*Weakened* mode on “Weakened" examples), users search less (73%) than they did when they didn’t expect the label to show up (82%).

of examples in which they might have expected the recommendation algorithm to surface the correct answer, they only conducted a further search 73% of the time, and as a result, achieved significantly worse total recall (average of 53%) than the other two modes (76%), as seen in Figure 4-1(a). They searched significantly less here than on the "difficult" examples ($p<0.05$, two-sided Mann-Whitney U test). This difference indicates that users were not solely searching further based on whether they thought a label was missing, but also based on whether they thought the recommendation algorithm should have been able to surface the correct label.

Therefore, on the examples where the *Weakened* users did not find a correct label in the recommendations, but expected to find one, they were less motivated to search. In these cases, they tended to accept substandard labels they perhaps wouldn’t have otherwise. Common failure modes include assuming there must be no matching label and (i) choosing that there is no label present (e.g., as a user did for `Diastolic CHF`), or (ii) settling for a related but suboptimal label (e.g., *ultrasonography* for `echo` instead of searching for *echocardiogram*). A *Weakened* user chose that there was no code for 4% of *Weakened* examples at median, and 6% on average.

This decreased search behavior leads to significantly worse results. A pairwise Mann-Whitney U test shows that while total recall was not significantly different between modes for “Easy" or “Difficult" examples, the “Weakened" mode was significantly worse at the “Weakened" examples than the other two modes ($p<0.02$ for both); see Figure 4-1(a).

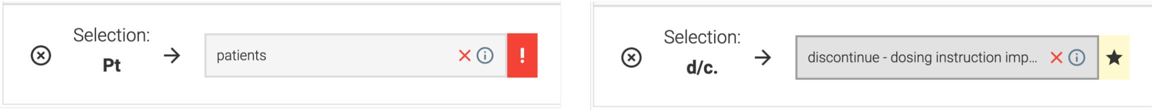


Figure 4-2: Flag indicators for suggestion confidence, displayed in the selection panel. Low-confidence (left) is indicated by an exclamation point on a red background, and high-confidence (right) is indicated by a star.

4.5 Stage 2: Annotation Suggestions

In this second stage of the user study, we study the effect of presenting a set of automatically pre-populated annotation suggestions, as described in Figure 3-1, that users can choose to accept, modify, or reject. Since there are existing systems that could pre-annotate at least half of the data, it has been suggested that starting from scratch might be an unnecessarily tedious exercise. However, there are fears users might lose engagement in the task and as a result, accept annotations with incorrect spans or incorrect labels, when models are imperfect. As a potential mitigation, we explore whether informing users that provided labels are high or low-confidence will make them more attentive when most necessary. An additional potential worry is a loss of agency; unlike in vanilla classification tasks, the user has to select which regions of the text to annotate. If the text comes partially pre-annotated, users might be less likely to take the initiative to annotate further. Both of these concerns could cause a feedback loop in which future models trained on this data become even more confident in error modes.

4.5.1 Experimental Design

The same 18 clinicians participated in this stage of the study, and they spent approximately three hours each on this stage of the study. They were now re-assigned randomly to one of four modes: the *No* suggestion mode (3 users), the *Standard* suggestion mode (4 users), the *Augmented* suggested mode (5 users), and the *Weakened* suggestion mode (5 users). All modes were provided with the automatic label recommendations from the *Standard* mode in Stage 1; the back-end search was also updated to that same algorithm, since we are no longer directly testing recommendation versus search.

The *Standard* suggestion mode displays suggestions for the examples where there was an exact match between the example text and one of the synonyms in the medical vocabulary (76% of the annotations in the gold standard). Of these suggestions, all spans and 85% of the labels are correct. The *Augmented* mode shows the exact same suggestions but includes a small flag on the label (see Figure 4-2), indicating whether the algorithm has high-confidence or low-confidence in the label. Approximately one-third of the suggested labels are considered low confidence; the high-confidence labels are 90% correct, and the low-confidence labels are 70% correct. The *Weakened* suggestion mode has the same setup as the *Standard* suggestion mode, but it also includes an additional set of 21 suggestions over incorrect spans. To mimic real-life algorithmic errors, these spans were taken from real incorrect span outputs of the clinical extraction systems cTAKES and MetaMap [15, 236].

Since users already had some exposure to the tool and annotation schema, they completed just four steps in the tutorial this stage, again receiving feedback iteratively after each step. The tutorial sentences were designed to contain span and label errors in the same proportion as the full study, so that users could understand the role of automated decision aid. As in the previous stage, they again labeled three notes each, which contained a total of 449 spans in the gold standard.

The evaluation is conducted as in the first stage, using the same metrics and comparing to the same gold standard. In this case, we also analyze the subsets of annotations across suggestion confidences and provided label and span accuracies. Where appropriate, we also compare our users' outcomes in Stage 2 to their corresponding outcomes in Stage 1, to understand individual shifts in behavior.

4.5.2 Results

Users were relatively accurate at assessing the correctness of the labels for pre-populated suggestions. Accuracy differed widely across users, but was not a function of skill.

When suggestion spans and labels were correct, users with suggestions accepted them over 99% of the time. The median user without suggestions had a 89% span recall and 86% total recall on this set of annotations, confirming that users without suggestions do miss some examples. When spans were correct and labels were incorrect, the median user accepted suggestions 17% of the time without modifications, and there were no observable difference between error rates between the modes (p=0.72, Kruskal-Wallis test). Namely, the presence of incorrect spans in *Weakened* mode did not appear to induce additional mistrust in the incorrect labels; users in the *Weakened* mode accepted incorrect labels at a similar 20% median rate to users in *Standard* and *Augmented.* Further, the presence of the confidence indicators in the *Augmented* mode did not make any noticeable impact on user's rate of modifying incorrectly suggested labels or their accuracy on the low-confidence subset. This matched their own feedback that "[the flags] didn't really affect the likelihood I accepted the suggestion" and that they do "not pay too much attention to the symbols."

Potentially due to the ease of accepting a pre-annotation in a single click, we observed a large difference between users in terms of how often they accepted incorrect label suggestions (a standard deviation of 0.14). While the vast majority of users accepted between 10 and 20% of incorrect suggestions, one user only accepted 8%, and another accepted 58%. The user who accepted 8% did not appear any better at the task; they had below-average total recall for their mode in Stage 1. Similarly, the user who accepted 58% did not appear less competent at the task; they achieved above-average accuracy for their mode in Stage 1. This suggests that our domain experts reacted differently to automation, but this reaction is not directly a function of skill. From our user survey, we do note that users' perceptions of annotation accuracy do not necessarily reflect true underlying accuracy; for example, one user reported that they found the pre-annotated labels to be 95% correct, despite their only accepting around 80%.

Though the majority of user errors were one-off, errors were not randomly distributed, and they indicated snap judgements based on concept name. For example, for a patient with a `persistent` cough, our system provided the incorrect label of *persistence*, defined as *mental perseverance*; this incorrect label was accepted by 10 users. For a patient suffering from `apical ballooning`, 6 users accepted the suggestion of *balloon dilatation*. While by name alone, the concepts sound like plausible labels, in both cases they were of the incorrect concept category (e.g., *balloon dilatation* is categorized as a *Procedure* instead of a *Problem*). While this category information is displayed to users, it seems they were not sufficiently engaged to utilize it. Our platform further provides a button for surfacing a label's definition, which was not taken advantage of by these users for either of these examples.

Users were slightly less accurate at getting rid of incorrect spans than incorrect labels.

On average, participants in the *Weakened* mode kept 33% of the 32 suggestions provided with incorrect spans, though the most conscientious user kept only 16%. Trust in incorrect spans was strongly correlated with trust in incorrect labels (ρ=0.70, Spearman). However, as before, it did not hold any significant correlation with users' span accuracy in Stage 1 (ρ=-0.29), indicating that trust may be independent of user's competence at the annotation task. For example, the user who only accepted 16% of incorrect spans had below average accuracy for their mode in Stage 1. Some of the provided incorrect spans included obviously incorrect selections such as `medical conditions. Exam` which spanned two sentences (kept by 2/5 users), or `of the superior segment branch`, which contained unnecessary prepositions (kept by 3/5 users). This indicates a decreased engagement. Other incorrect spans indicated concrete concepts, but were not clinical in nature, e.g., `sister`, and are therefore not supposed to be tagged. No users in any other mode chose to annotate any of extraneous spans mentioned here.

	Stage 1 Nontrivial Span Recall	**Stage 2 Nontrivial Span Recall**	**User Survey on the Process of Adding New Annotations**
User A	63%	37%	“I made sure to double check if there were parts that were not annotated."
User B	71%	56%	“I reviewed sections just in case I missed some... but the marked sections were fairly comprehensive."
User C	68%	58%	“Having pre-suggested parts actually made it easier to scan the remaining unmarked parts for words to annotate"
User D	63%	62%	“[Pre-annotations] definitely freed up mental bandwidth to allow me to spend more energy on the unmarked text."

Table 4.2: Differences for four representative users between nontrivial span recall on Stage 1 (where they had no pre-annotations) and Stage 2 (where they did). Nontrivial span recall calculates the proportion of spans that users took the initiative to annotate that would not have been pre-annotated by our annotation suggestion algorithm. While the drop between Stage 1 and Stage 2 recall indicates that users took less initiative in practice, users believed they were more thorough in their exit surveys.

Users with pre-populated suggestions exercised less agency in creating new annotations, but they noted the opposite in their exit surveys.

While users with pre-populated suggestions overall had slightly higher total recall than users without, they initiated the creation of fewer additional annotations. If we consider the subset of nontrivial annotations—annotations where suggestions were not provided and are therefore the most important to annotate—users with suggestions annotated fewer of these spans (median of 58%) than users without suggestions (median of 74%). Most striking was when we contrasted users to their own nontrivial recall from Stage 1; as a note, in Stage 1, there was no significant difference between nontrivial recall between the modes ($p>0.4$, two-sided Kruskal-Wallis). While Stage 1 did not contain suggestions, for a direct comparison, we restrict to the set of nontrivial annotations that *would not have had* suggestions, had we applied the same suggestion algorithm. Users without suggestions increased on average 4% between the two stages, indicating Stage 2 may have been slightly easier, but users with suggestions dropped on average 12%. In an Aligned Rank transform test with factors of (i) stage and (ii) whether users received pre-annotations, there was a statistically significant effect of stage ($p<0.005$) and of the interaction between stage and mode ($p=0.01$).

Only a single user had a higher nontrivial span recall in the presence of suggestions. The difference was most pronounced in compound terms, where users without suggestions had a median of 64% nontrivial span recall on compound terms, and users with suggestions only had a median of 31%. For example, all users without suggestions tagged `RV thrombus` (compared to 46% of users with suggestions) and `decompensated CHF` (compared to 50% of users with suggestions).

Once again, this loss of agency was not correlated with users' prior accuracy and, qualitatively, users did not recognize it was happening. While a user's nontrivial recall results at Stage 1 and Stage 2 were highly correlated (Spearman's $\rho = 0.76$, $p<0.001$), there was no correlation between their loss in agency (measured as difference in nontrivial recalls between stages) and their prior performance (Spearman's $\rho = 0.25$. $p>0.3$). When prompted in their exit surveys, all stated that they believed the pre-annotations made it easier to tag the nontrivial spans. We display a few representative responses in Table 4.2, alongside their drop in nontrivial span recall. Even users (e.g., User C, User D) who were confident that they had greater bandwidth and performed better displayed a small drop in performance.

Users were faster on annotations that came with pre-populated suggestions, but when labels were incorrect, they were slower than the users annotating from scratch. There was also a correlation between faster speeds and more errors.

Label time was calculated as the time between highlight and label choice (for annotations without suggestions) and time between a suggestion click and label choice (for annotations with suggestions). 76% of all annotation spans in the gold standard were suggested to users as pre-annotations. Of these, 81% had both a correct span and label. In the cases where both span and label were correct, users in suggestion modes only needed to verify the label was correct, and they were on average 30% faster at label selection than users not in suggestion modes (a median of 5 vs 7 seconds). However, when labels are incorrect, users with suggestions were on average 5 seconds slower than users annotating from scratch; users also spent an average of 5 seconds on suggestions over incorrect spans. On examples without pre-annotations, users annotate at approximately the same speed as they had on comparative examples in Stage 1. We find that in our context the efficiency gains are slim in our context and depend greatly on their underlying accuracy.

While users lauded the efficiency gains, we find that users with very high efficiency gains tended to be more inaccurate. In practice, some speedy users would quickly accept pre-annotations in less than a second; video revealed that their primary tactic was to first quickly deal with pre-annotations, and then go back and scan through the remaining text. In contrast, the user with the second lowest rate of accepting erroneous label suggestions was also one of the slowest, taking almost twice as long as the median. In exit surveys, users believed the pre-annotations were great for efficiency, noting that they were "much faster," "more efficient," and "cut down time significantly." Empirically, however, their utility is less clear-cut.

4.6 Discussion

The results of our study suggest implications for the appropriate amount of automation to include in our and similar platforms. We found that our domain experts were relatively adept at dealing with incorrect labels: both searching for additional labels when recommendations failed, and modifying incorrect labels in pre-populated annotations. This would indicate that there is minimal downside to the inclusion of label recommendations in the platform UI, regardless of whether it is presented in a list format, or as a single prediction. In contrast, in Stage 2, users were slightly less likely to get rid of incorrect spans in pre-populated annotations, even when spans were clearly incorrect. This would indicate that the set of pre-populated annotations would need to prioritize high precision (surfaced annotations correspond to true spans) over high recall (most gold standard annotations are surfaced). Additionally, we found the presence of pre-populated annotations redirected users' attention and caused a loss of agency, and such techniques should be employed with caution.

Further, our study revealed some characteristics of our domain experts. First, they did not realize they had ceded agency, and this underscores that they are not fully cognizant of their interactions with automation. Therefore, their own qualitative conclusions on how they operate in teams are generally insufficient; this aligns with recent work from Buçinca and Lin et al. [37]. Second, we found that the negative impacts of automation were user-specific and correlated across aid modalities. However, these impacts were independent of a user's competency at the task.

The optimal decision regarding how to integrate decision support into a system may depend on the downstream use case. Counter to prior concerns, we found that decision support increased users' average recall. Therefore, if the output of a task is being directly used for decision making, then the full inclusion of decision support may be useful. However, if the output of a task is to be used for further model training, there would be a possible fear of creating a feedback loop with pre-populated annotations, since we found that users are less likely to annotate concepts that the machine missed. When fed back into models, users' output may erroneously confirm the machine's decision.

The clinical text annotation task involves a rich set of subtasks and decisions that allowed us to probe questions of trust and agency in expert decision makers. However, our findings do not necessarily extend to simpler settings. For example, due to our large potential label space, it was implicitly clear to users that the model is presenting only a subset of possibilities. In contrast, if a domain expert only has to make a decision with a binary outcome (e.g., *Does a pathology image indicate cancer?*), the user may develop a different dynamic with decision support, since the problem is more constrained. That being said, binary problems like image classification often involve smaller implicit subtasks (e.g., *Does this patch of the image display cancer?*). As automated methods attempt to focus users' attention on the subtasks they deem most relevant, our findings on agency may still apply. On the flip side, there are far more complex forms of clinical decision support (e.g.. AI-assisted documentation) that require additional study, extending the initial foray of Li et al. [151].

Other limitations include the low-stakes setting of our study task. Since we did not deploy in a live setting, and there was no consequence for incorrect actions, users may not have been as careful as they would have been under greater pressure. Additionally, we found that presenting model confidence had little-to-no-impact, and users admitted to not using the confidence flags. Other UI techniques might have made these confidences more salient to users.

Several techniques could mitigate the shortcomings of human-AI teams from a system design perspective, both for our task and more broadly. For example, we could withold pre-annotated suggestions during the initial round of annotation; after, a second annotator could incorporate any missing spans from pre-annotated suggestions. This may dampen the effect of the observed loss of agency. We could also experiment with pre-populated span suggestions without pre-selected labels, which could force users to remain critically engaged. Another option would be to continually engage the user with regular feedback; currently, they only receive direct feedback during the tutorial phase. Possibilities include (i) purposefully planting a fraction of erroneous recommendations and suggestions and alerting the user to when they've accepted one and (ii) alerting the user when they skip over a span they should have annotated. Additionally, the amount of training may affect outcomes, and we could investigate whether more rounds of early tutorial feedback would lead users to better mentally characterize the shortcomings of the decision support.

Further, we found that automation had widely varying impacts on users and ceded agency tended to correlate across modalities. One simple option would be to filter users with high misplaced trust in automation or to predict when user log data indicate they may be running on autopilot. However, that is not always a realistic solution, and our work further found that susceptibility was not correlated with competency or skill. Therefore, another future direction would be to understand users' susceptibilities by their early results (e.g., in the tutorial) and then adjust the level of automated decision aid provided accordingly. This could involve adapting notions of *learning to defer* [165, 215, 127] or *learning to complement* [300] from the pure classification setting to the label selection setting. Under these paradigms, the decision support itself could be adjusted to the strengths of users, rather than being optimized in isolation.

Instead of just attempting to fix the automation-induced noise at the point-of-annotation, another path would be to design machine learning algorithms that anticipate and adapt to the noisy data. Natarajan et al. showed that in the presence of noise, binary machine learning classifiers can still be successfully trained, if the patterns of noise are well-characterized and below random (e.g., there is an estimate of what fraction of the time a specific outcome may be incorrect) [186]. In these cases, models can be trained by re-weighting their objective function by a factor dependent on the probability of user error. In our case, we would need to adapt such algorithms to account for probabilities of users accepting incorrect spans or skipping certain categories of annotations, for example. Given our observed empirical results, we posit that machine learning methods designed to overcome the pitfalls of human-AI teams are an important area of future study [218].

4.7 Conclusions

In this chapter, we studied the impact of decision aid on domain experts via empirical lab studies on clinicians (n=18) over extended periods of use of PRAnCER. More broadly, our platform enables efficient annotation of text documents and could help scale data set creation in a domain where annotated data set sizes have been historically small.

On the whole, we found that our domain experts remained appropriately skeptical of label recommendations, and they formed an intuition for when further searching was required. Similarly, they mostly recognized when pre-populated labels were incorrect. As a result, the introduction of automatic label recommendations is unlikely to lead to significant bias. Unfortunately, our domain experts do fall susceptible to handing over agency to algorithms. Without them realizing it, the presence of pre-populated suggestions leads them to lose critical engagement in the task and add fewer new annotations than each had previously. Given that these new annotations are the ones that provide us the most new signal for training models, we would be in danger of models being hampered in their training process.

As forms of automation like AI-assisted writing become incorporated into more and more decision processes, it becomes paramount for us to understand how automation affects expert decision makers. As we found with varied susceptibility among users, issues of algorithmic trust and agency extend far past user confidence and expertise to a more intrinsic behavior. As we do in this work, understanding, characterizing, and quantifying that behavior in complex, real-world tasks is an important first step. It informs the design of both user interfaces and machine learning systems that can optimally combine the strengths of humans and AI and mitigate their joint shortcomings.

Chapter 5

Human-AI Framework for Clinical Timeline Creation

Acknowledgement of Co-authors I would like to acknowledge my undergraduate mentee and co-first author Jason Zhao, who conducted the software development for the framework described in this chapter.

5.1 Introduction

In this chapter, I move from token-level annotation to note-level annotation. We now explore a joint human-AI framework for the annotation of temporal clinical events in notes (e.g., date of metastatic recurrence), towards the creation of clinical timelines. The annotation of clinically meaningful events often involves an arduous and expensive process of manual abstraction by domain experts [12]. This manual chart review process can be particularly time-consuming when studying chronic diseases (e.g., cancer), in which patients have lengthy clinical timelines to sift through and structure [26]. While understudied, the extraction of temporal elements enables researchers to ask key questions, such as whether treatments extend life span, and how disease progression trajectories differ across subpopulations [19].

The field of clinical information extraction aims to circumvent the arduous manual abstraction process via automatic systems that leverage natural language processing

(NLP) and more recently, deep learning in particular [42, 287, 217]. While these systems can improve efficiency, this efficiency often comes at the expense of reliability. Due to the complexities in healthcare, such models remain imperfect, and their deployment is therefore subject to skepticism [234]. On the other extreme, relying on manual effort alone is often not a scalable option for running large real-world evidence studies.

In high-risk settings such as healthcare, one solution to this trade-off is to complement machine learning models with expert human aid that can step in when models fail [113]. In this chapter, we integrate humans into the loop by letting our extraction model, if needed, iteratively query an expert; this process is shown in Figure 7-1. By optimizing the queries solicited, we aim to reduce model errors while preserving most of the time and cost savings that automated NLP systems provide.

In Figure 5-1, we show the outcome of running our system over three patients who experience metastatic recurrence. The left two plots display two patients' clinical timelines; since pathology reports directly indicate a metastatic diagnosis, our system felt sufficiently confident to directly extract date of metastatic recurrence. The rightmost plot displays another patient's timeline where the system first queried a label to gain confidence.

Our framework for human-guided search helps regulate the efficiency-accuracy tradeoff for event identification from sequences, and we introduce the metric *Model-derived Query Utility* to choose the optimal query. Over a cohort of breast cancer patients, we empirically show our system's efficacy on identifying date of (i) metastatic recurrence and (ii) the start of a therapy regimen, two tasks that are crucial to leverage oncology real-world evidence.

5.2 Related Work

One classic way of controlling the accuracy-efficiency tradeoff is via rejection learning. In rejection learning, a system learns a classification model h and a rejection model r. The rejection model r may decide to either "reject" a data point x and incur a cost $c(x)$

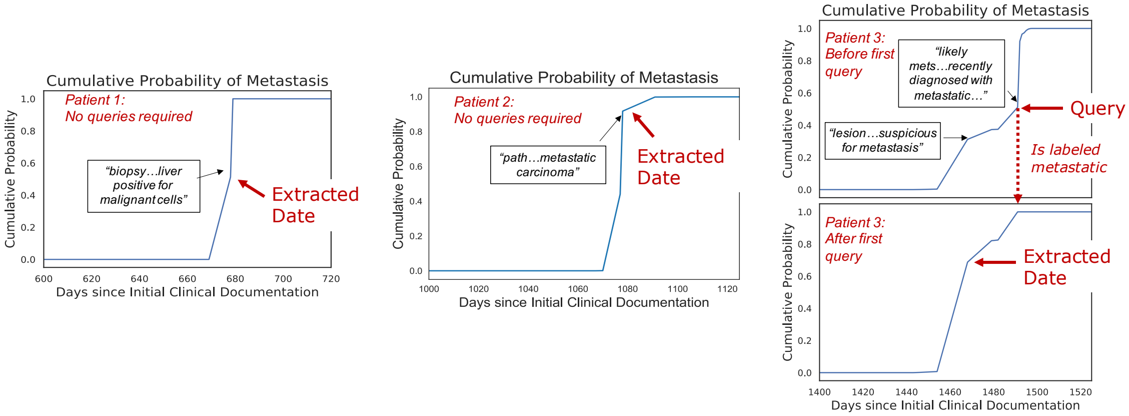


Figure 5-1: Plots of the model's cumulative probability over time that a given patient was metastatic. For patients 1 and 2 on the left, due to the sudden jump in cumulative probability, the model feels sufficiently confident to directly extract the date without eliciting any human input. For patient 3, on the right, before any queries are made, there is more ambiguity as to the date of metastatic recurrence. However, after a single query at the marked index, the model calculates a new posterior probability and is now sufficiently confident to extract.

(which can be viewed as asking the human expert to make the prediction), or decide to predict using h and incur a cost corresponding to misclassification error. Learning to defer [165, 83, 181] builds on the rejection learning framework by additionally allowing the system to adapt to different types of experts, where the cost of “rejection" also depends on the expert prediction m, i.e., $c(y, x, m)$. Unlike learning to defer, our formulation does not attempt to model different classes of experts, and instead assumes that experts are oracles for the labels. In contrast to this prior work however, we extend the notion by considering deferral specifically for just a single label in a sequence.

While sequences have not been studied in the context of the rejection learning framework, there have been sequence-based strategies in the active learning setting, in which the algorithm chooses full sequences to be labeled as additional training data [245]. Classic approaches include uncertainty-based methods, that measure average or total label entropy, and disagreement-based methods; these approaches have been primarily studied over entire sequences, and where the labeling budget is based on the number of sequences, not factoring in variable time required per sequence. Tomanek and Hahn [270] looked at per-instance labeling, instead of full sequences, for a token-labeling task; their method involved querying any record where the marginal probability of its likeliest label was under a given threshold. However, in our settings, labels in a patient timeline are highly correlated, so this solution may not be optimal. Fang et al. [67] used a reinforcement learning approach for sequence selection, but assumed a large amount of available labels for validation, often impractical in healthcare settings. Furthermore, we note the underlying purpose of selecting labels in active learning is different than in rejection learning. In active learning, the goal is to find those sequences most informative for training a new model for a downstream evaluation, which is not necessarily equivalent to identifying outliers or incorrect sequences. Finally, our objective function is tailored to event identification, as compared to generic sequences.

Practically, there is great utility in efficient extraction of clinical fields from free-text notes. Clinical information extraction is an active subfield, mining diverse variables from comorbidities to treatment exposures to adverse events [287]. Due to the clinical importance of recognizing metastasis to oncology cohort creation, there have been multiple studies showing that one can effectively extract metastatic status from a set of aggregated patient notes [155, 28]. While such studies have been able to accurately identify metastatic status at a patient-level, they have not focused on the timing of metastatic recurrence, crucial clinically to assess outcomes using real-world evidence. As a bridge towards temporal precision, Banerjee et al. [19] worked on identifying whether metastatic recurrence was present within a given quarter, by aggregating notes across 3-month time spans. Carrell et al. [42] worked on a broader task to identify occurrences of *any* breast cancer recurrence (ipsilateral, regional, or metastatic), but discussed how their error rates have implications for the potential introduction of bias. Our second task, the extraction of timing of oral cancer therapy start, has been explored with both rule-based and ML-forward implementations [285, 3]. However, in both tasks, no research has studied how to improve extraction by adding a human-in-the-loop. Here we work towards more-fine grained temporal accuracy and allow users to set their own tolerance for permissible errors.

5.3 Methods

In this section, we explain our human-guided search framework, consisting of the event identification task, our event extraction model, and our algorithm *Model-derived Query Utility* for choosing a query for a human-in-the-loop. Our framework iteratively decides between using the extraction model directly and querying an expert.

5.3.1 Preliminaries

Our dataset is composed of n sequences denoted by $(X_1, \cdots, X_n)$ where each sequence X_i is a set of a variable number of records t_i: $X_i = (X_i^1, \cdots, X_i^{t_i})$. We associate with each record X_i^j a timestamp which we store in the list T_i, i.e., record X_i^j occurs at timetstamp $T_i[j]$. Finally, denote y_i to to be the index at which the event of interest occurs for sequence i. From this dataset, we construct a set of latent sequence labels z_i^j for $j \in \{1, \cdots, t_i\}$, where

$$z_i^j = \begin{cases} 0 \text{ if } j < y_i \\ 1 \text{ if } j \geq y_i \end{cases} \tag{5.1}$$

This framework is sufficient for modeling a variety of clinical temporal extraction tasks. For example, in our first application, X_i represents the sequence of notes for patient i, y_i represents the index of metastatic recurrence for that patient, $T_i[y_i]$ represents the date of that recurrence, and $z_i^j = 0$ for the clinical notes before a patient's recurrence, and $z_i^j = 1$ for the clinical notes after.

5.3.2 Event Extraction Model

Given a sequence $X_i = (X_i^1, X_i^2, X_i^{t_i})$, we would like to extract the event index $y_i \in \{1, 2, \cdots, t_i\}$. We tackle this task by training a model q_θ to directly fit the distribution of y_i given the sequence X_i. In other words, $q_\theta\left(y_i = j | X\right)$ is the probability density function for the model's belief that the event occurs at index j. We parameterize

q_θ using an LSTM [111], a Recurrent Neural Network. Our extraction model takes as input the embeddings for each note, X_i, and passes it through a 1-layer bidirectional LSTM to obtain the hidden states h_i at every timestep. Finally, the h_i are passed through a fully-connected layer followed by a softmax to obtain the model probabilities $q_\theta\left(y_i = j|X\right)$ at every timestep i. Further details on embeddings are located in the experiments.

We then define $p_\theta\left(z_i^j = 1|X\right)$ to be the cumulative probability function that the event has occurred by index j for sequence i, formally defined below:

$$p_\theta\left(z_i^j = 1|X_i\right) = \sum_{0 \leq r <= j} q_\theta\left(y_i = r|X_i\right). \tag{5.2}$$

The extraction model then estimates the event index, which we call $\hat{y}$. We define $\hat{y}$ to be the index at the median of the cumulative event distribution, namely the j such that $p_\theta(z_i^j = 1|X) \geq 0.5$ and $p_\theta(z_i^{j-1} = 1|X) < 0.5$. Recall that the timestamps are stored in T_i, so that the estimated event timestamp is $T_i[\hat{y}]$.

5.3.3 Human-guided extraction

We now introduce a human expert who we assume can accurately label each record X_i^j with its label z_i^j to reduce ambiguity in our search space; the described process is illustrated in Figure 7-1.

In the case with a single event of interest, the feedback received from the expert can be sufficiently described by two variables: let a be the largest labeled index with a 0 label, and let b be the smallest labeled index with a 1 label. The index a is initialized at 0, and b is initialized at $t_i + 1$. We can now update our cumulative probability for the event occurrence in terms of this additional input, namely:

$$p_\theta\left(z_i^j = 1|X_i, a, b\right) = \frac{\sum_{a \leq r \leq j} q_\theta(y_i = r|X)}{\sum_{a \leq r \leq b} q_\theta(y_i = r|X)}. \tag{5.3}$$

We define $\hat{y}_i(a, b)$ to be the estimated index of the occurrence given the obtained bounds a and b. The objective of our system is to now iteratively select the query whose labeling would have the greatest effect on shifting this estimate, up until the estimate is sufficiently stable. If a note j is labeled, the bounds will update to some $(a^{'}, b^{'})$ —either (a, j) or (j, b) depending on the label of j—which has the potential to change our estimate from Equation 5.3. Our system chooses the index, which if labeled, would shift the estimate date by the largest number of days. We formalize this notion as *Model-derived Query Utility* $\mathbb{E}[\Delta_i^j]$, which is defined in terms of the current estimate $\hat{y}_i(a, b)$:

$$\begin{aligned} \mathbb{E}[\Delta_i^j] &= p_\theta\left(z_i^j = 0|X_i, a, b\right) \cdot |T_i\left[\hat{y}_i(a, b)\right] - T_i\left[\hat{y}_i(j, b)\right]| \\ &+ p_\theta\left(z_i^j = 1|X_i, a, b\right) \cdot |T_i\left[\hat{y}_i(a, b)\right] - T_i\left[\hat{y}_i(a, j)\right]| \end{aligned} \tag{5.4}$$

Given this metric, at each iteration, we select:

$$\arg\max_{j \in [a,b]} \mathbb{E}[\Delta_i^j] \tag{5.5}$$

label the record at index j, and update $[a, b]$ to $[a^{'}, b^{'}]$ accordingly. We continue iteratively until $\mathbb{E}[\Delta_i^j] < L$ for a hyperparameter L that controls the accuracy-efficiency tradeoff. We note that if $\mathbb{E}[\Delta_i^j] < L$ at the first iteration, no queries are conducted at all.

We prove in Appendix B.2 that in a zero-information scenario with a uniform distribution, this formulation collapses down to binary search, which has optimal time complexity for search. This is a greedy approach, but we also devised a reinforcement learning approach for learning a querying strategy, which we compare to in Section 5.5 and describe in Appendix B.3.

5.4 Data

5.4.1 Cohort

We considered a retrospective cohort of breast cancer patients who presented to Memorial Sloan Kettering Cancer Center. All patient records were de-identified of

PHI (e.g., names, dates), both within structured fields and unstructured clinical notes. This research was reviewed and determined to be IRB-exempt.

Over this cohort of breast cancer patients, we evaluate our system on two clinically important extraction tasks, each described in further detail below. All variables were extracted from notes by non-clinician abstractors who specialized in breast cancer clinical data collection. Quality control of labeled variables was ensured via scheduled auditing reports by an overseeing management team.

5.4.2 Metastatic Recurrence

Our first task is identification of the date of metastatic recurrence, defined as spread of the disease to distant organs or occurrence of unresectable locally advanced disease. The date of metastatic recurrence was abstracted as the date of pathologic confirmation, if available. When an initial metastatic biopsy was not performed, the date was extracted based on radiologic recurrence instead.

Existing literature has shown that simply identifying whether a patient has experienced a metastatic recurrence (without time localization) is relatively solvable for machine learning classifiers [155, 28]. Therefore, for our training and evaluation, we restrict our cohort to only those patients who experienced metastatic recurrence.

In addition to a cohort of 476 patients where we have exact extracted date of metastasis, we additionally have a group of 379 patients with labels with less temporal specificity; the date of metastasis is approximated as the date of first line metastatic therapy. While not used for evaluation, we use this approximate cohort to train the extraction model. A consort diagram detailing the cohort creation process is available in Appendix B.1.

5.4.3 Therapy Start

The second evaluation task we consider is date of therapy start. In particular, we evaluate on three drugs taken orally: tamoxifen, letrozole, and palbociclib. Due to their method of administration, such drugs appear less consistently in structured data (e.g., compared to intravenous chemotherapy), and therapy regimens may be shifted from the original prescription time due to delays in insurance or pharmacy pickup.

For patients in our breast cohort, abstractors structured all drugs (oral and intravenous) taken for their breast cancer treatment, including those drugs prescribed for the patient's course at another institution. Therefore, we restrict our cohort for this task accordingly to those patient-drug pairs which are feasibly recoverable, given that our data set does not include scanned records from outside practices and all dates are de-identified within the notes themselves.

In our training cohort, we exclude examples in which the drug administration preceded the first record at the institution or the drug was not mentioned in any notes within one month of the noted start date. For our evaluation cohort, we further excluded examples in which the drug was not mentioned anywhere on the abstracted start date or in the two weeks following, and where there was no follow-up within 2 months; the purpose was to exclude patients who come in only for a second opinion or primarily for surgery.

The literature has shown high accuracy at the binary task of determining whether a patient has taken a certain drug [3]. Therefore, for this task, we assume we are given a patient and a drug they took, and are asked to return the initial date of therapy start. Training was conducted over 8,843 patient-drug pairs, validation over 1890, and evaluation over 508 patient-drug pairs. Training and validation occurred over all drugs, whereas evaluation took place over just the three aforementioned oral drugs. A full consort diagram is present in Appendix B.1.

5.5 Experiments

In this section, we quantify the efficacy of our human-guided extraction framework on real-world extraction tasks. Label queries were solicited synthetically from “experts", i.e., we make the framework assumption that domain experts can accurately conduct the extraction task and can return the true z_i^j given X_i^j.

5.5.1 Comparisons

First, we compare our human-guided extraction framework to model-only and human-only baselines. Additionally, within our framework, we consider other objective functions for selecting a query in addition to the *Model-derived Query Utility* method described in 3.3. To our knowledge, there is not previous work that has tackled this human-assisted sequential formulation, so besides the deferral of whole sequences, these objectives are also novel. Each is described below:

- In *Extraction Model Only*, we deploy the extraction model with no human input and estimate $\hat{y}_i$ directly.

- In *Vanilla Binary Search*, we do not use the extraction model and estimate how long it would take a human using binary search to pinpoint the timing of the clinical event.

- In *Whole Sequence Deferral*, we choose the $P\%$ of notes with highest label entropy [245] to undergo a full labeling (via binary search). As in the previous example P is tested over a variety of hyperparameter choices.

- With *Policy Model*, we follow the framework from Fang et al. [67]; we devise our own parametrization and define the reward for a given query as the number of days closer the estimate is after a query. We vary hyperparameter C, the cost for querying an expert which is reflected in the reward function. We define a policy model whose action space consists of querying for a label or directly predicting. Full details of our implementation of this method can be found in Appendix B.3.

- We additionally devised a new query method *Model-Augmented Binary Search*. In it, the domain expert is iteratively queried at the first record j for which the cumulative probability of the event occurrence $p_\theta(z_i^j)$ is at least 0.5. The model can terminate its search early if the left and right bounds for the timing of the event drops below D days, where D is tuned as a hyperparameter.

5.5.2 Metastatic Recurrence

Task Setup

Next, we evaluated our system on the metastasis extraction task. We first trained our extraction model on a dataset of 693 patients, 379 with approximate labels of metastasis (based on date of first metastatic therapy), and 323 with gold extracted dates. The extraction model was first pre-trained on the gold patients, and then run on the full training cohort. 50 additional gold labels were reserved for validation of the extraction network, and 103 were used for testing the final system. For the reinforcement learning comparison, we use 192 of the patients allocated for training/validation of the extraction network for training and validation of the policy network instead.

Since documentation of some patients' metastasis occurred outside this cancer center, we conducted evaluation based on a shifted metastasis date—namely, the first note in our available records in which the patient was confirmed metastatic, and the closest possible we could get on our data set, given that dates in text have been de-identified. For this task, we considered notes across clinical oncology, pathology, and radiology. In our test set, patients had a median of 80 notes each.

Implementation Details

First, each record was encoded in a bag-of-words (BOW) fashion using n-grams ($1 \leq n \leq 3$) that occurred in at least 2% of notes. To generate a lower-dimensional embedding, we trained a LASSO regression to predict the latent z labels from individual notes, and then included only the 238 features (8% of the original total) with a nonzero weight in the LASSO regression in our final BOW embedding. These features included “mets", “to bone", and (stage) “IV". We note that in our preliminary experiments, we found that these BOW embeddings outperformed more complex note embeddings, generated via word2vec or convolutional neural networks.

We use these reduced BOW embeddings per note as input to our extraction model q_θ, in which the output dimension of the LSTM is size 64. We train our network for 20 epochs using a batch size of 8. We use the Adam optimizer [130] with initial

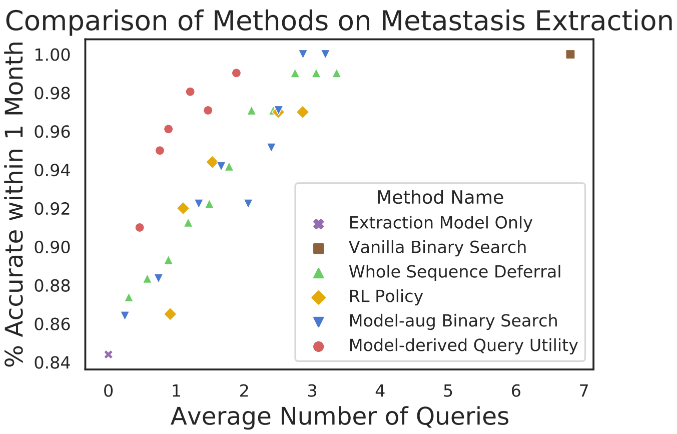


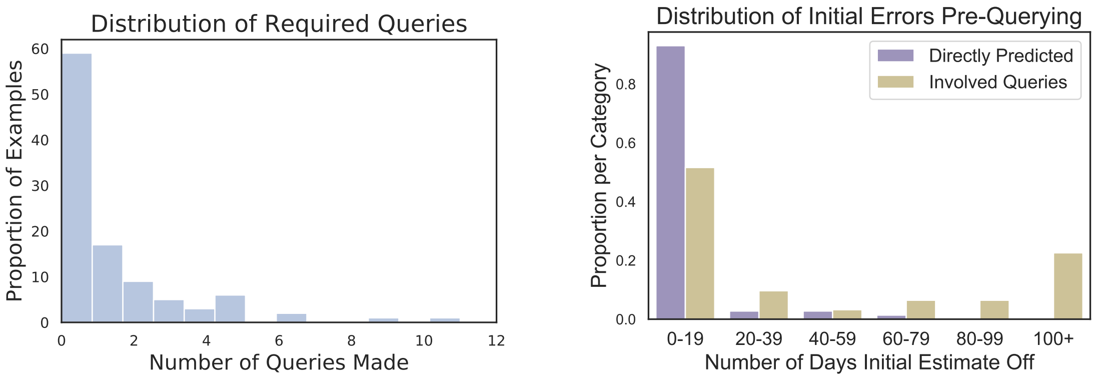
Figure 5-2: The above plot visualizes the trade-off between the average number of queries per patient, and the percentage of patients whose estimated date of metastatic recurrence fell within one month of the first metastatic note. At the bottom left, one can see the accuracy if the extraction model was used alone, and on the top right, the number of queries necessary if binary search was used to pinpoint each variable. We can observe that employment of *Model-derived Query Utility* provides the optimal trade-off, compared to the other methods.

learning rate 0.01 and train using L_2 regularization with a coefficient of 0.001. These hyperparameters were selected using the best-performing model on the validation set.

Results

The extraction model alone pinpoints the correct record indicating metastasis within 2 weeks 73% of the time, within a month 84% of the time, and within two months 88% of the time. If we examine the errors, they often arise when there is confusion in the original diagnosis, e.g., a lung or breast metastasis that may be a second primary, or from conflicting information in the original note, e.g., due to copy-forwarding. An example of the latter can be seen in Figure 5-1, where a note said both that the patient had “likely metastasis" and was “newly diagnosed metastatic", conflicting signals that a human is better suited at parsing. A fully manual binary search approach requires an average of 6.8 queries per patient.

Results on the metastasis task after adding a human-in-the-loop are in Figure 5-2; for methods with hyperparameters that tune the efficiency-accuracy tradeoff, outcomes are displayed over a variety of hyperparameters. The ideal case is to be in the top



(a) alttext (b) alttext

Figure 5-3: The left plot shows the distribution of number of queries solicited under the *Model-derived Query Utility* algorithm at L=1, in which the correct date was pinpointed within a month 98% of the time; 57% of patients required no queries at all. The right plot shows the distribution of initial errors of the extraction model, split into (i) the patients whose dates were directly extracted and (ii) the patients for whom queries were requested. We note that there were minimal errors on the set not queried.

left corner (full accuracy with no queries required). We can see in the plot that the *Model-derived Query Utility* method is Pareto optimal on this dataset. Compared to the other methods, approximately one fewer query is needed per patient on average to achieve the same accuracy.

The left of Figure 5-3a) displays the distribution of required queries for the *Model-derived Query Utility* method, with the hyperparameter $L = 1$. Under this setting, 98% of metastases are correctly localized within a month and 99% are correctly localized within 2 months. For approximately 60% of patients, no queries are required at all, and for approximately another 20% of patients, only a single query is needed. On the right in Figure 5-3b), we can see the distribution of errors of the estimates $\hat{y}$ from the initial extraction model, before any querying. We split our distribution into the 60% of patients that were directly extracted and the 40% of patients that required further querying. For the directly extracted patients, we notice that a large majority have very close initial estimates; this is a desirable property, because our model can in fact achieve good performance on these patients without any queries. On the other hand, for patients which *Model-derived Query Utility* decided to undergo at least a single round of human-guided querying, initial predictions are far more erroneous, sometimes having over a 100 day difference, validating the importance of having a human-in-the-loop.

We examine cases in which our method required no queries while the comparison baselines required 3 to 4 queries. In one such example, there was large spacing between the radiologic evidence plus pathologic confirmation, and oncologist follow-up. The extraction model waited until the oncologist follow-up to become fully confident. The *Model-derived Query Utility* model was sufficiently confident that querying in this case was unnecessary, but due to the larger time gap, the other methods queried regardless.

5.5.3 Therapy Start

Task Setup

We first trained the extraction model on the 8,843 patient-drug pairs from breast cancer patients and validated on 1,890. For the RL policy model, we reduced the train and validation set size for the extraction model by 3022 and 483 pairs respectively to use in training and validating the policy network instead. The final system was tested on the 508 pairs in the evaluation set. Since a patient could take multiple drugs and therefore be in several pairs, train/validation/test sets were created to ensure no patient overlap between sets.

For this task, we only considered notes from clinical oncology; since we restrict to notes mentioning the drug, the patients in our evaluation cohort had only a median of 14 notes each. Evaluation is conducted on the basis of the ground truth abstracted date, even if it does not correspond to a note.

Implementation Details

Due to the large signal-to-noise ratio in records, we preprocess records based on whether they contain a mention of the drug; a mention of a drug is a string match of the brand name, generic name, or a common abbreviation (e.g., “palbo" for palbociclib). We tokenize into sentences and remove all sentences that do not mention the drug;

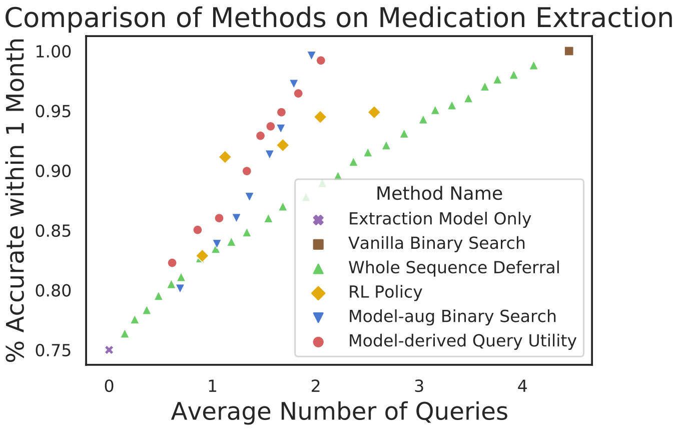


Figure 5-4: Results for the medication extraction task. The plot above shows the trade-off between the average number of queries per patient, and the percentage of patients that fall within one month of the gold standard therapy start dates. *Model-augmented binary search* and *Model-derived Query Utility* perform comparably and provide the most consistent performance across hyperparameter choices.

notes with no mention of the drug were removed. Moreover, each mention of the target drug was replaced by a universal `CURR_DRUG_TOKEN`, and mentions of other common breast cancer medications were replaced by a universal `OTHER_DRUG_TOKEN` to allow for generalizability of features across drugs.

Then, similar to the previous task, each preprocessed record was encoded in a bag-of-words (BOW) fashion using n-grams ($1 \leq n \leq 3$), and thresholded for a vocabulary size of 650. Unlike the previous task, we omit the use of LASSO regression to generate a lower-dimensional embedding, based on results of initial experimentation.

Analogous to the previous task, we parametrize q_θ using an identical architecture, a bidirectional LSTM with output dimension 64, followed by a fully-connected layer. We train our model for 5 epochs using the Adam optimizer and an initial learning rate of 1e-3. We train using a batch size of 8, and we select the best model using early stoppage by taking the best-performing model on the validation set.

Results

The extraction model alone gets the correct date within one month 74% of the time and within two months 90% of the time. If we examine where errors are made, ambiguity in the underlying extraction model often arises from conflicting reports between prescriber's plans and patients' actions. For example, a note may indicate that the patient is "starting the drug" whereas the next note includes that the patient has "refused to switch treatment." In examining these cases, we do note poorer calibration of our extraction model, where the model tends to be overconfident; in the previous example, the cumulative probability was high once the doctor stated the patient's regimen had started, despite the downstream later evidence they had not yet begun. We also ran a binary search baseline with no model input, which required an average of 4.4 steps to finish at completion.

Results factoring in human input for start of oral therapy extraction are displayed in Figure 5-4. On this dataset, *Whole Sequence Deferral* based on label entropy performs by far the worst of all the methods here, nearing random performance. The RL policy model does relatively well at selection of the highest yield queries, but performance quickly tapers off. We find that our cumulative probability model-based approaches, *Model-derived Query Utility* and *Model-augmented Binary Search*, strongly outperformed other baselines and performed equivalently to one another. For example, we can achieve 90% accuracy within a month (a 60% error reduction) with only 30% of the queries a full search would require.

5.6 Discussion

Our results show that a small amount of human oversight is often sufficient to increase the reliability of one's model outputs, validating our human-guided framework for event extraction. Using the *Model-derived Query Utility* method, with fewer than an average of a single query per patient, dates of metastatic recurrence were correctly recovered within 2 months for 98% of patients in the test set, compared to 88% with the extraction model alone. This new accuracy is sufficiently high enough for most clinical research, while the initial model accuracy alone may have incurred worries about potential bias and noise trickling into the downstream applications. Moreover, this approach only requires 13% of the annotation effort that a binary search approach would have required. This indicates that there is great promise in using a joint extraction process with *Model-derived Query Utility* to manage the trade-off between effort and accuracy.

Compared to metastasis, the wins are less stark for extraction of medication therapy date. We partially attribute this to our pre-processing, which led to a shorter timeline length. In our pre-processing, we had already filtered out any notes that do not directly mention the drug, since (i) such notes are unlikely to contain the start date, and (ii) they would violate the assumption that a domain expert could tell the status of the drug regimen based on the note alone. However, such pre-filtering is not necessarily typical in clinical abstraction settings, so true time savings over a manual chart review may be larger than our results may indicate. We found that a further detriment to the medication extraction model was a more miscalibrated extraction model than the one for metastasis, in which the output probabilities did not fully reflect the true probability of misclassification error. This skewed the query utility downwards due to model overconfidence. Therefore, recalibration of extraction models may be a useful intermediate step, using existing off-the-shelf techniques [97].

Future Work Our existing framework can directly extend to jointly extract multiple events in a clinical timeline, assuming their relative ordering is known. For example, one may want to track monotonic disease staging across time, e.g., when cancer progresses from Stage n to $n+1$. q_θ would transition to a multivariate model, and $\mathbb{E}[\Delta_i^j]$ could be redefined as the sum of the expected date shifts for each new stage. Evaluation on such a dataset remains a direction for future study.

Another direction for future work is to increase the granularity of a note presented to the human labeler, by showing or highlighting just a specific subportion of a note. Due to practices like copy-forwarding, notes can become bloated; clinical oncology notes contain a median of over five hundred tokens in our dataset. Therefore, there is utility not just in localizing notes temporally, but also indicating which portion of the note to focus on. A model could learn to imitate what experts looked at in practice or learn to highlight in a fully unsupervised fashion. Other methods like summarization or presentation of a few salient points are also ripe for future study.

Finally, this chapter currently does not make use of structured data in the EHR, but an interesting direction for further work would be to learn weak signals automatically (e.g., once a patient has a certain ICD code, they almost certainly are metastatic).

Limitations As is always the case, we made modeling assumptions that while generally reasonable, may be simplifications of the messiness of real world clinical data. For example, our approach of labeling z_i^j hinges on disease stage being monotonically increasing; while metastatic breast cancer is uncurable, there are other diseases one might want to label with non-monotonic disease staging. Additionally, there are many events like cancer progression that can occur at multiple points in a patient's timeline, for which the modeling assumptions do not hold. In the following chapter, we suggest a new method that fills this gap for disease progression.

Additionally, we assumed that at any given note, it would be possible to tell whether or not an event of interest had already occurred. This is often a sound conclusion, since oncology notes often contain a summary of the patient's disease and treatment course thus far, due to copy-forwarding and note bloat. In our drug start date experiment, we only included those notes that specifically mention the drug, to ensure this assumption upheld. However, that assumption may not necessarily hold true across clinical specialties or note writing styles.

Another limitation is that our evaluation was run as a simulation of human-in-the-loop interaction, but not as an actual user study. A real user study would allow us to quantify the speed and the workflow of a labeler before and after use of our system. Further, a user study may reveal real-world preferences that may inform tweaks to the reward functions in our system. As a potential example, while manual chart review often involves jumping through a patient's timeline, it may be unnecessarily cognitively complex in this setting. Instead, it may be useful to incur a penalty if notes queried for labels are out-of-order.

Here we also presume the extraction of a single event type at a time. In reality, styles and schemas differ, and important future work involves jointly optimizing for different events simultaneously—both mathematically and in terms of human interaction. Further, this chapter relied on a fair amount of labeled data to kickstart the joint human-AI annotation, and as the number of events increase, this may not remain realistic. In the following chapter, we explore ways to make an initial model more label-efficient.

5.7 Conclusions

We have introduced a framework for a human-in-the-loop system that regulates the efficiency-accuracy trade-off for event identification in clinical timelines. We have contributed a *Model-derived Query Utility* metric for query selection that consistently performs as well or better than other metrics across hyperparameter settings on two clinical event identification tasks: (i) metastatic recurrence and (ii) the start of an oral therapy regimen, two tasks that are important to oncological research. Further, we are the first to show that *rejection learning* can be used effectively on temporal, sequential data, which saves valuable domain expert annotation time in the clinical setting. Our framework can help enable institutions to leverage the real-world evidence in their unstructured EHR notes at scale, enabling cohort creation and retrospective clinical studies that would may otherwise have been prohibitively tedious or expensive to conduct.

Chapter 6

Self-supervision for Longitudinal Clinical Text

Acknowledgement of Co-authors I would like to acknowledge my co-first author, Hunter Lang, who provided the theoretical grounding (not fully included in this thesis) for the self-supervised objective described in this chapter. Hunter was additionally heavily involved in the experimental design of both the synthetic and real-world experiments.

6.1 Introduction

Typically, limited *labeled* data is available for downstream clinical information extraction tasks of interest, and labels can be prohibitively expensive to obtain [29, 311]. Fortunately, given the large amount of unlabeled data, *self-supervision* is a promising avenue. In self-supervision, models are first pre-trained to optimize an objective over unlabeled data, with the goal of learning representations that capture important semantic structure about the input data modality. For example, in masked language modeling for text, the model is trained to predict the identity of randomly masked tokens. Performing well at this objective should require a representation of sentence syntax and semantics. Once pre-trained, self-supervised representations can be used for downstream supervised tasks.

However, despite the success of self-supervision across domains, the development of new self-supervised objectives has been a largely heuristic endeavor. *Why* does pre-training improve performance on downstream tasks? Whether this happens depends on both the self-supervised objective and the downstream task itself. But the assumptions linking the self-supervised objective to the downstream tasks of interest are rarely, if ever, made explicit.

In this chapter, we design a self-supervised objective with a *particular class* of data distributions and downstream tasks in mind. We aim to make explicit the type of distributions and downstream tasks on which we expect this method to work. We are interested primarily in the types of time-series that arise in longitudinal health data for patients, particularly those with chronic conditions, e.g., cancer, autoimmune disorders, and neurodegenerative diseases. These time-series include long sequences of clinical notes, insurance claims data, biomarker measurements, or combinations thereof. A key differentiating feature of these data is that a given trajectory can *change quickly*. For example, a patient may develop a new symptom between subsequent healthcare visits, or a certain biomarker value (e.g., blood pressure) may dramatically increase.

Additionally, these changes largely tend to be *irreversible* with respect to time. For example, once the word "metastasis" appears in a clinical note, nearly all subsequent notes tend to comment on the state of that metastasis (so the word "metastasis" appears in those notes as well). To train a good representation for certain downstream problems (e.g., "what is the patient's current disease state?"), the self-supervised objective should *attend* to these changes, rather than suppress them.

These properties make such data distributions unsuitable for several existing self-supervised objectives. For example, Franceschi et al. [75] train a model so that the representation of each time segment is more similar to those of its subsegments than the representation of a randomly chosen segment from another trajectory. Similar techniques have been used to learn image representations from video: two subsequent video frames are likely to contain the same objects [175, 92]. These approaches are all similar to the idea of *slow feature analysis* [302] for extracting representations of an

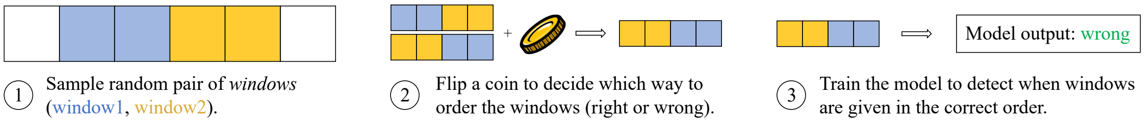

+
Model output: wrong
① Sample random pair of *windows* (window1, window2).
② Flip a coin to decide which way to order the windows (right or wrong).
③ Train the model to detect when windows are given in the correct order.

Figure 6-1: Depiction of the data-generation and learning process for order-contrastive pre-training. For each trajectory, a pair of consecutive windows is sampled uniformly at random, flipped with probability 0.5, and presented to the model. The model is trained to predict whether the presented pair of windows is in the correct (+1) or incorrect (−1) order.

input signal that change slowly over time. Representation learning techniques based on the ideas of slow feature analysis are appropriate for some downstream tasks and time-series data types, such as the ones studied in the works above, but not, we argue, for data where the latent variable of interest (such as disease state) can have large changes between subsequent time steps (e.g., between visits to a medical care center). Motivated by the example of chronic diseases in healthcare, we focus on the setting where time-irreversible features are highly useful for downstream classification, and where these features may exhibit large changes between subsequent time steps.

In this chapter, we introduce a self-supervised objective called *order-contrastive pre-training* (OCP). For each trajectory in the input data, we sample random pairs of time segments, switch the order for half of them, and train a model to predict whether a given pair is in the correct order (*positives*) or in the incorrect order (*negatives*).

This procedure is shown in Figure 6-1. OCP is very similar to an existing technique known as *permutation-contrastive learning*, or PCL [116]. PCL was also designed to take advantage of temporal dependence between features of the input signal to learn useful representations. The key difference between these two objectives is in the sampling of the *negatives*. Where the negatives in OCP are incorrectly-ordered window pairs, the negatives in PCL are *random* window pairs from the same trajectory, and could be in the correct order. In their simplest forms, the positive samples for the two methods are identical: pairs of consecutive windows in the correct order.

Intuitively, the same time-irreversible features that are useful for the OCP and PCL objectives should also be useful for downstream prediction tasks. To formalize and quantify this, we study a class of data distributions motivated by the preceding discussion. When the representation belongs to a simple hypothesis class (effectively, when the representation is a *feature selector*), we prove a finite-sample bound on the *downstream* error of a representation learned using OCP. Although this setting is much simpler than those that appear in similar work (it involves *linear*, rather than nonlinear, representations of the input data), we show that this model still admits interesting behavior. In particular, we give an example of a data distribution in this setup where OCP and PCL provably learn different representations. Additionally, this model indicates that even when two methods have the same performance with infinite unlabeled data, there is an unlabeled-sample-complexity benefit to using a "clean" distribution of negatives, which matches well with prior work on other contrastive learning algorithms [52].

We supplement this motivating theoretical study with experiments on real-world time-series data. Our results indicate that for the types of data and tasks discussed above, both OCP and PCL representations can enjoy better downstream prediction performance than those trained using existing self-supervised baselines. Moreover, complementing our theoretical results, we show a real-world scenario where OCP outperforms PCL in the low labeled-data regime despite the seemingly minor difference between the two objectives. Given that OCP and PCL only differ slightly in their negative sampling, these results give further theoretical and empirical evidence for the importance of the negative sampling details in contrastive learning, complementing several recent works [52, 224, 158].

6.2 Related work

Order pre-training. Others have found order-based self-supervision useful for more complex time-series data, but without theoretical study. For example, learning the order of frames within a video yields representations useful for downstream activity classification [70, 174, 145, 291]. Most similar to our work, Hyvärinen and Morioka [116] introduced *permutation-contrastive learning* (PCL) and proved *nonlinear* identifiability for representations learned using PCL in an ICA setting. That is, they gave distributional conditions where PCL provably recovers the "correct" nonlinear representation of the input given infinite unlabeled data. Our theoretical and empirical results indicate that there can be nontrivial differences between OCP and PCL's downstream performance. Deeper understanding of what data distributions and downstream tasks are "right" for PCL versus OCP (and for other contrastive sampling methods) is an interesting direction for future theoretical study.

Pre-training for medical time-series. Several other pre-training objectives have been explored on clinical time-series data. A contrastive learning setup similar to our *patient-contrastive* baseline has shown promising results on electrocardiograph signals [63, 131]. Banville et al. [22] studied a contrastive objective for electroencephalography signals, in which windows of a signal are judged to be similar if they occur within a certain time gap, and dissimilar if they are far away in time. Intuitively, this objective is well-suited to data where the true representation "changes slowly" with time, as with Franceschi et al. [75] (discussed in Section 6.1). Other objectives include auto-encoding [74] and masked prediction over text and tabular data, to mixed results [259, 115, 317, 168]. Multi-task pre-training supplies improvements, but unlike our work, it relies on additional labeled data from closely-related downstream tasks [168].

Self-supervision theory. Like our work, Saunshi et al. [235], Liu et al. [158] and Tosh et al. [271, 272] give downstream finite-sample error bounds for representations learned using particular contrastive learning objectives. Our motivating theoretical setting and proof techniques are simpler than the ones considered in these works, but we show that our setup in Section 6.4 is (i) complex enough to allow for some of the same nontrivial behavior observed by contrastive methods in practice (Sections 6.4.1, 6.5.1) and (ii) it has some practical applications (Section 6.5.2).

6.3 Order-pretraining algorithm

We suppose each data point X is a time series, $X = (X^1, \ldots, X^\tau)$, where τ is the number of sample points and may vary with X. We also suppose the samples take values in some common set $\mathcal{X}$. Let a window w be an element of $\{1,\ldots,\tau\}$, and let X^w be the corresponding element of X.[1]

Given a trajectory X, we use the following generative process to sample a data point (Z, Y) for our contrastive task. First, Y is chosen uniformly at random from $\{-1, 1\}$. Next, random windows W and W' are chosen (in a manner explained below). The segments X^W and $X^{W'}$ corresponding to windows W, W' are combined into a tuple Z. The pair (Z, Y) is then a sample for the contrastive task. A model h, given by a composition of a classifier $c \in \mathcal{C}$ and a representation $g \in \mathcal{G}$, is trained to predict Y from Z:

$$\underset{h=(c,g)}{\text{minimize}}\, R_{ord}(h) := \mathbb{E}_{(Z,Y)}[h(Z) \neq Y] \tag{6.1}$$

Here $h(Z) = h(X^W, X^{W'}) = c(g(X^W), g(X^{W'}))$. That is, h first computes the representation g for each window, then uses a classifier c to predict whether the tuple Z is in the correct order. The representation g can then be re-used on a downstream task. The remaining design choice is to specify the process for sampling windows.

Order-contrastive pre-training. A simple choice for sampling random windows W, W' is to sample a random pair $(W, W+1)$ in the *correct* order when $Y = 1$, and $(W+1, W)$ in the *incorrect* order when $Y = -1$. We refer to the optimization problem (6.1) with this choice of sampling as *order-contrastive pre-training* (OCP). This can easily be generalized to non-consecutive window pairs. The pretraining task (6.1) is thus to *contrast* windows in the correct order with windows in the incorrect order.

Permutation-contrastive learning. Another simple choice is to again sample a pair $(W, W+1)$ in the correct order when $Y = 1$, but sample a *random pair* when $Y = -1$. This is the data generation process for *permutation-contrastive learning* [116]. Note that the only differences between OCP and PCL are that in PCL, (i) the negative samples ($Y = -1$) need not be consecutive, and (ii) some are in the correct order. The distributions of positive samples are identical. We refer to the

---

[1] For simplicity, we only consider windows of size 1. Our results straightforwardly generalize to windows of arbitrary size ℓ, where $w = (w_1, \ldots, w_\ell)$ is a subinterval of $\{1, \ldots, \tau\}$ and $X^w = (X^{w_1}, \ldots, X^{w_\ell})$.

procedure (6.1) with this sampling as *permutation-contrastive learning* (PCL). This exactly matches the contrastive sample distribution in Hyvärinen and Morioka [116, equations (10)-(11)]. Here, the pretraining task is to contrast consecutive windows in the correct order versus random window pairs.

Comparison. These two sampling methods seem very similar—they only differ slightly in the distribution of negatives (i.e., conditioned on $Y = -1$). However, we show theoretically and empirically in the following sections that they can learn very different representations when used in (6.1), and they can have different unlabeled sample complexities even if they eventually find the same representation. This gives further evidence of the importance of negative sampling for contrastive learning methods (see, e.g., Chuang et al. [52]). We give a finite-sample bound for the downstream classification performance of a representation learned using OCP in a simple setup motivated by time series data and predictive tasks in healthcare.

6.4 Finite-sample guarantee for time-irreversible features

In this section, we study a class of distributions motivated by applications to time-series data in healthcare. We assume for simplicity that each $X^t \in \mathcal{X} = \{0,1\}^d$. We identify a set of four assumptions for which we can prove a finite sample guarantee for the set of *feature selector* representations $\mathcal{G}$. Here we use $\mathcal{F}$ to refer to the downstream hypothesis class, and we overload Y to refer to the *downstream* label of interest.

Assumption 1. *There exists a set S of time-irreversible features. Formally, $\forall i \in S, \forall t$, $\mathbb{P}[X_i^t = 1, X_i^{t+1} = 0] = 0$.*

Assumption 2. *When the features in S are not changing, the other features are time-reversible. More formally, for all t, and all $v, v' \in \{0,1\}^d$, if $v_S = v'_S$, $\mathbb{P}[X^t = v, X^{t+1} = v'] = \mathbb{P}[X^t = v', X^{t+1} = v]$.*

Assumption 3. *There are no "redundant" features in* S*. For all* U *such that* $S \not\subset U$*, there exists* t *and* $v \in \{0,1\}^d$*,* $v' \in \{0,1\}^d$*, with* $v_{U\cap S} = v'_{U\cap S}$ *and:*

$$\begin{aligned} &\min\mathbb{P}[X_S^t \subsetneq X_S^{t+1}, X_U^t = v_U, X_U^{t+1} = v'_U];\\ &\quad\mathbb{P}[X_S^t \subsetneq X_S^{t+1}, X_U^t = v'_U, X_U^{t+1} = v_U]) > 0. \end{aligned}$$

Assumption 4. *The features* S *are suitable for the downstream classification task (here* Y *refers to the downstream label):*

$$\arg\min_{f\in\mathcal{F}} \mathbb{P}[f(X_S) \neq Y] = \arg\min_{f\in\mathcal{F}, g\in\mathcal{G}} \mathbb{P}[(f \circ g)(X) \neq Y]$$

Intuitively, the first two assumptions guarantee that the feature-set S is an optimal choice of representation for the OCP pretraining objective (when $\mathcal{G}$ is the class of feature selector representations), and the third (more technical) assumption guarantees that the optimum is unique (e.g., by preventing the possibility that pretraining leaves out a feature in S that is redundant for the ordering objective, but useful downstream). The last assumption ensures that the features S are suitable for the downstream classification task on the population: the loss achievable by the best $f \in \mathcal{F}$ using X_S as the representation is the same as the loss achievable by using the best (f, g) pair.

When these assumptions are satisfied, we prove a finite-sample bound for a model pretrained using OCP. The bound only depends on the VC-dimension of the *downstream* hypothesis class, $\mathrm{VC}(\mathcal{F})$, rather than on $\mathrm{VC}(\mathcal{F} \times \mathcal{G})$.

Like some results in the nonlinear ICA literature (e.g., Hyvärinen and Morioka [116]) our results only apply in the regime where there is enough unlabeled data to identify the "correct" representation. It's then immediate that only $\mathcal{F}$ factors in to the labeled-data dependence. However, our model also allows us to give upper bounds on the *amount* of unlabeled data required to reach that regime. This allows us to more rigorously study other aspects of contrastive pretraining, such the role of *bias* in the negative distribution, which has been shown to affect the performance of other contrastive learning algorithms [52].

We now give a simple example of a class of distributions satisfying these assumptions, grounded in our running application of health time-series data. Despite its simplicity, our findings suggest that this model allows for several interesting phenomena that also occur in practice, which could make it useful for further study of contrastive learning methods on time-series.

6.4.1 Extraction example

A common task in clinical informatics is to extract for each time t the patient's structured disease stage, which enables downstream clinical research [124, 125]. Each time point X^t could be an encoding of the clinical note from a patient's visit at time t. Let $Y^t \in \{0, 1\}$ be the observed label for time point X^t. The end goal is to train a model f over a representation g to minimize the downstream risk:

$$\underset{(f,g)}{\text{minimize}}\, R(f, g) := \mathbb{E}_{(X^T,Y^T)}[f(g(X^T)) \neq Y^T].$$

Here we make a prediction for every time point, and the expectation is over the time index T as well as the trajectory (X, Y).

Model. For each $A \subset [d]$, we denote by X_A^t the random variable corresponding to indices A at time t. Suppose the set of feature indices $[d]$ is partitioned into three types of features:

- A set $S \subset [d]$ of time-irreversible features. We also assume that each $i \in S$ has a nonzero probability of activating on its own, without the other features in S. That is, for each $i \in S$ there exists t with $\mathbb{P}[X_i^t = 0, X_i^{t+1} = 1, X_{S\backslash\{i\}}^t = X_{S\backslash\{i\}}^{t+1}] > 0$. This ensures that assumption 3 is satisfied. Such features include the onset/progression of chronic conditions and markers of aging [209]. For example, appearance of the word "metastasis" in a clinical note.
- Noisy versions $\hat{S}$ of S: for each $j \in \hat{S}$, there exists $i \in S$ with $\mathbb{P}[X_j^t = X_i^t] = (1 - \epsilon_i)$, with $\epsilon_i > 0$, for all t. Additionally, X_j^t is conditionally independent of the other variables (for all times) given its parent variable X_i^t. For example,

the presence of certain interactions with the health system—such as deciding to attend physical therapy—may be a noisy reflection of the patient's true disease state, which is captured by X_S^t.

- Background, reversible features B: features such that for all t and all $v, v' \in \{0,1\}^d$,

$$\begin{aligned}&\mathbb{P}\left[(X_B^t, X_{[d]\setminus B}^t) = (v_B, v_{[d]\setminus B}),\right.\\ &\qquad\left.(X_B^{t+1}, X_{[d]\setminus B}^{t+1}) = (v'_B, v'_{[d]\setminus B})\right] =\\ &\mathbb{P}\left[(X_B^t, X_{[d]\setminus B}^t) = (v'_B, v_{[d]\setminus B}),\right.\\ &\qquad\left.(X_B^{t+1}, X_{[d]\setminus B}^{t+1}) = (v_B, v'_{[d]\setminus B})\right]\end{aligned}$$

  Consider, for example, common words such as "and", "chart", etc., in a clinical note, whose presence or absence gives no order information.

Note that we *do not* make any independence assumptions between the features in this example other than the ones mentioned above.

In this section we gave a simple example of a class of distributions, together with an *assumption* linking the distribution to the downstream task (Assumption 4—the time-irreversible features are the most useful ones for downstream classification) for which we can *prove* that OCP gives a more parsimonious bound on the labeled sample complexity. While the example in Section 6.4.1 seems straightforward, we show now that it still admits interesting behavior. In particular, there are distributions that satisfy Assumptions 1-3, but where PCL and OCP learn different representations.

PCL versus OCP: different infinite-data optima. There are examples of the model from Section 6.4.1 where PCL and OCP learn provably different representations even with infinite unlabeled samples, despite the minor difference in their sampling schemes.[2] Intuitively, the existence of a periodic feature (such as a procedure always

---

[2] The example we use includes *nonstationary* features, which violates the assumptions under which PCL is proven in Hyvärinen and Morioka [116] to find the "right" representation, so this does not contradict those results.

performed at a particular time of day) is strongly predictive of whether two samples are consecutive, but need not be predictive of whether a pair of consecutive samples are in the correct order. Concretely, consider a feature X_i such that $X_i^{t+1} = 1 - X_i^t$, and $X_i^1 \sim \text{Ber}(0.5)$. Inclusion of this feature doesn't violate Assumptions 1-3—indeed, X_i would qualify as a "background" feature under our model—so the theorem described in [6] guarantees that OCP finds the correct representation. However, in PCL, every non-consecutive sample is a negative. But only non-consecutive samples can have $X_i^t = X_i^{t'}$, so X_i^t is helpful for the PCL objective. We treat this example more formally in the full manuscript [6], but our synthetic results in Section 6.5 also show that a background periodic feature can affect the PCL representation.

"Debiased" negatives. PCL has some negatives that are actually in the correct order. Prior work on contrastive learning has called this "bias" in the negative distribution [52]. What's the role of this "bias?" Does it affect the learned representations? Does it affect the amount of *unlabeled* data required to find a good representation? For distributions satisfying assumptions 1-3 and when $\mathcal{G}$ is the class of feature-selectors, we answer these questions in the negative and positive, respectively.

In particular, consider the analogue of OCP that instead of *always* choosing $(W+1, W)$ when $Y = -1$ (Y as used in OCP, not the downstream label), instead just chooses a random pair (W, W') with $|W - W'| = 1$ (i.e., a random consecutive pair). We refer to this as OCP-biased, since some of the negatives are actually in the correct order. However, the following theorem shows the estimator obtained by minimizing this objective is *not* biased in a statistical sense:

Theorem 1 (informal)**.** *When assumptions 1-3 are satisfied, S is also the unique optimal representation for OCP-biased.*

However, it *does* affect the bound on unlabeled sample complexity required to obtain a good representation:

Proposition 1 (informal)**.** *The upper bound on the sample complexity required for OCP-biased to identify S is worse than the upper bound for OCP.*

While this proposition only compares upper bounds, our synthetic experiments (see Figure 6-2) indicate that OCP is more sample-efficient than OCP-biased. We prove these results in the full manuscript [5].

6.5 Experiments

6.5.1 Synthetic data

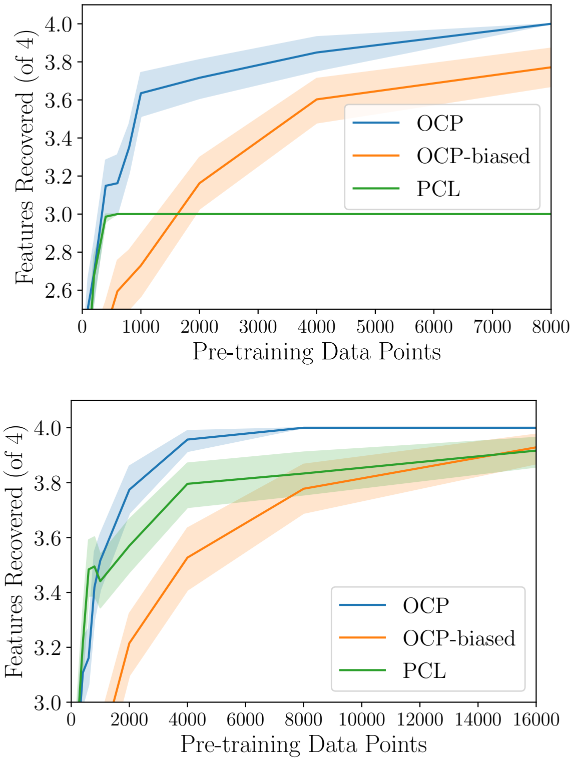


Figure 6-2: Synthetic experiments admitting different behavior across pre-training setups. (Top): PCL is unable to ever recover all four true features in S; (Bottom): PCL recovers the true representation, but requires higher sample complexity than OCP.

We demonstrate the importance of negative sampling over two synthetic datasets from the model in Section 6.4. Each distribution contains $|S| = 4$ alongside a number of noisy features. We generate pre-training datasets of different sizes (50 to 16,000) and sample pairs from each dataset according to OCP, PCL, and OCP-biased. We then conduct a logistic regression with L_0 penalty over the sample pairs and analyze how many variables in S were correctly recovered. The top panel of Figure 6-2 shows a distribution where PCL does not recover S in the infinite data limit—this distribution includes a periodic background feature in B that is selected by PCL. The bottom panel of Figure 6-2 shows a distribution where PCL is able to recover all of S, but requires a larger sample complexity than OCP. In both cases, OCP and OCP-biased find the same representation, but the former has better dependence on unlabeled data. We provide the details and explanations for these experiments in Appendix C.1.

6.5.2 Real-world data

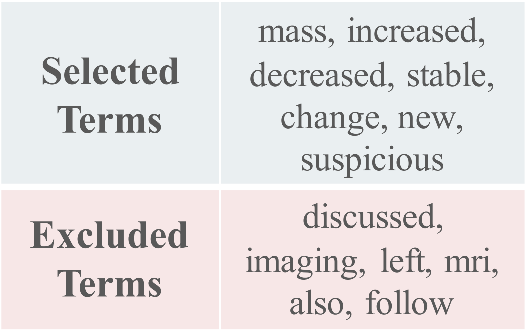
We show OCP yields significant improvements in the low-label regime on extraction from clinical notes.

Progression dataset. We utilize a dataset of fully de-identified clinical notes from Memorial Sloan Kettering Cancer Center. This research was reviewed by the MIT Committee on the Use of Humans as Experimental Subjects and determined to be IRB-exempt. The dataset contains data for 82,839 patients with cancer, with a median of 12 radiology notes each. Each radiology note focuses on one body area (e.g., chest CT scan). In addition, we have a subset of 135 patients with progressive lung cancer with 1095 labeled radiology notes. Each note was labeled post-hoc by a dedicated thoracic oncologist as 'indicating progression' (19%), 'not indicating progression' (79.5%), or 'ambiguous' (1.5%).

Experimental setup. We investigate extraction of these binary progression labels from the *Impression* section of the note. The labeled data was split via 5-fold cross-validation: each fold contained sets of sizes 64% (train), 16% (validation), and 20% (test); for a given fold, no patient examples were ever split between sets. On each fold, we used the test set to benchmark models trained using different amounts of the labeled training data: from just 5 training patients ($\frac{1}{16}$) to all of the training patients. We excluded patients with downstream labels from pretraining. For contrastive pre-training schemes, a pretraining window pair was sampled once per each unique body

	Fraction of training data				
Available features	**1**	**1/2**	**1/4**	**1/8**	**1/16**
OCP subset	0.864	0.860	0.847	0.808	0.786
All features	0.856	0.851	0.818	0.723	0.726
Most common	0.767	0.767	0.728	0.687	0.658
Random subset	0.740	0.747	0.727	0.639	0.634

(a) Mean note-level AUC of regularized logistic regression over different dataset sizes. Averaged over the 5 folds, performance was optimal for each dataset size when restricted to the features with nonzero coefficients recovered by OCP.



Selected Terms	mass, increased, decreased, stable, change, new, suspicious
Excluded Terms	discussed, imaging, left, mri, also, follow

(b) Example features that OCP selected (top) or excluded (bottom) for downstream prediction.

Figure 6-3: Linear representation space experiment to validate assumptions of our model apply to real-world data. Quantitatively, we find downstream wins from restricting the model feature space to those found useful for the order-contrastive task. Qualitatively, the features important for the order pre-training are the same we would expect to be useful for the downstream extraction task.

area (e.g., chest, brain) that was scanned at least twice, capped at five locations per patient. This resulted in ≈158,000 samples for pretraining.

Pre-training for feature selection. We first validate our modeling assumptions from from Section 6.4 using a linear model. The goal of this section is to roughly validate our assumptions and the setup of our theoretical model. We compare downstream progression extraction performance of (i) a vanilla logistic regression model and (ii) a logistic regression model only using the features selected by OCP. We test on all five folds for five training dataset fractions.

For each experiment, our dataset is featurized using the unigrams and bigrams that occur in at least 5% of the labeled training data set. They are vectorized using the *term frequency-inverse document frequency* weighting scheme, via scikit-learn [205,

Table 6.1: Performance of deep methods on cancer progression extraction. The first row contains the mean AUC of OCP ± its std dev. The following rows contain the mean AUC advantage of OCP over each comparison method, and the percentage of time OCP outperforms that method, across the 3 seeds and 5 folds.

	Fraction of training data				
AUC diff. (OCP Win %)	**1**	**1/2**	**1/4**	**1/8**	**1/16**
OCP AUC	0.87 ± .03	0.86 ± .04	0.84 ± .04	0.82 ± .03	0.81 ± .03
OCP – BERT	0.08 (93%)	0.12 (100%)	0.12 (100%)	0.18 (100%)	0.22 (100%)
OCP – FT LM	0.03 (80%)	0.04 (82%)	0.04 (82%)	0.08 (93%)	0.10 (89%)
OCP – Pt-Contrastive	0.03 (86%)	0.03 (77%)	0.05 (91%)	0.09 (91%)	0.12 (97%)
OCP – PCL	0.00 (53%)	0.00 (46%)	0.03 (64%)	0.03 (76%)	0.06 (87%)

BSD 3-clause license]. We conduct feature selection as an optional intermediate step preceding progression extraction. For OCP, we train a logistic regression model with L_1 penalty over the 158,000 pre-training pairs of consecutive radiology notes. The regularization constant was set such that there were 50 ± 5 features with nonzero weights. In addition to OCP-derived features, we select the 50 most common features, and 5 random subsets of 50 features to serve as a comparison.

We train scikit-learn logistic regression models for downstream progression extraction over each feature set; further details are in Appendix C.2. Results can be seen in Figure 3a. Even with a simple bag-of-words representation, feature selection with OCP outperforms *directly* training a tuned logistic regression model on the available labeled data ("All features"), especially for small dataset sizes. A paired t-test finds that the model with OCP-selected features is significantly better than the direct-downstream model on a sixteenth of the data ($p < 0.05$). Note that selecting the most common features or a random set of features does not compare, showing that OCP does not improve performance by simply reducing the feature dimension in a redundant space.

We manually examined the OCP-selected features and their coefficients (Figure 6-3(b)). The features included (e.g., *increased, decreased*) strongly indicate disease progression, while those discarded (e.g., *discussed*) largely seem to be noise. Of the nonzero coefficients, 76% have a positive weight; this indicates that the pre-training model focuses mostly on features that have been *turned on* to conduct the ordering task, fitting with our motivating theoretical setting.

Nonlinear representations. We now study the use of OCP for pre-training nonlinear representations. We compare performance of a BERT model pre-trained using OCP to several other self-supervision methods. We investigate the BERT base model and the BERT base model after it is pre-trained using: (i) *FT LM*: fine-tuned masked language modeling over an equivalent number of impressions, (ii) *Pt-Contrastive*: a patient-level contrastive objective (identical positive sampling to OCP and PCL, but each negative is a random note of the same note type from a *different patient*, similar to Diamant et al. [63]), (iii) *PCL*: contrastive pre-training with PCL sampling (each negative is a random pair of notes of the same type from the same patient), (iv) *OCP*: contrastive pre-training with OCP sampling (each negative is a pair of notes of the same type in the incorrect order). All pre-training is conducted over three seeds, and all three contrastive objectives were trained with the same number of pairs (158,000). Implementation for language modeling and contrastive pre-training came from Wolf et al. [305, Apache-2.0 License] with full details in Appendix C.2. After model pre-training/fine-tuning, the self-supervised representation layers were frozen, and a single L_2-regularized linear layer was added on top. The goal of freezing was to isolate the effect of pre-training to understand representation quality, due to the instability of training BERT on small downstream tasks [322].

Results can be seen in Table 6.1. The top row shows that OCP has only a modest drop in performance even when trained on the data from just 5 patients ($\frac{1}{16}$). Since correlations exist in AUC across the 5 folds, 3 seeds, and 5 dataset sizes, standard statistical comparison testing is inappropriate. Instead, we present the mean increase in AUC from OCP, as well as the percentage of the time OCP outperformed the comparisons. Unsurprisingly, BERT alone (trained on non-clinical text) unsurprisingly does not perform well out-of-the-box; fine-tuning with language modeling improves performance, but still suffers in the low data regime. Among the contrastive objectives, the cross-patient objective is the weakest, which may follow since its pre-training task was the easiest (82% accuracy on validation). It could rely on features that differed between patients, instead of being forced to focus on the temporal features that differed *within* a patient's timeline. PCL performs equivalently to OCP at large data sizes, but at the smaller data set sizes, it loses to OCP a large majority of the time.

6.6 Limitations and Conclusion

We have shown both theoretically and empirically that order-contrastive pre-training is an effective self-supervised method for certain types of time-series data and downstream tasks. On real-world longitudinal health data, we find that representations from OCP significantly outperform others, including the similar PCL, in the small data regime. Concretely, being able to structure variables from longitudinal, label-scarce, data in health records could enable us to evaluate large scale retrospective datasets and potentially inform future clinical trials and patient care.

However, OCP is not always suitable. For example, cases of temporal leakage (e.g., the date in a note) can lead to weak OCP (and PCL) representations downstream, since they provide a shortcut during pre-training. While dates are straightforward to censor, more complex global nonstationarities irrelevant to downstream tasks would present a challenge to these methods. We additionally wish to emphasize that not all clinical tasks have time-irreversible expressions in the data (e.g., acute/temporary conditions, such as pregnancy), so the motivating model assumptions (particularly Assumption 4) should be considered before applying OCP.

Our theoretical setup and results in Section 6.4 also serve to highlight that contrastive pre-training methods can be very sensitive to the precise sampling details, and provide a simple model for studying these details that is still complex enough to capture some empirical phenomena. The comparison to PCL captures the importance of sampling *hard* negative samples, as also found by Robinson et al. [223]. Our initial experimentation found this as well, as distinguishing between consecutive notes during pre-training was found to be more effective than distinguishing between further pairs of notes. This suggests that obtaining broader theoretical guidelines for selecting a contrastive distribution is an interesting direction for future work. Future work could also involve augmenting the objective to other tasks that involve the metadata behind notes, such as predicting not just the ordering of the notes, but the time gap between a pair of notes as well.

Chapter 7

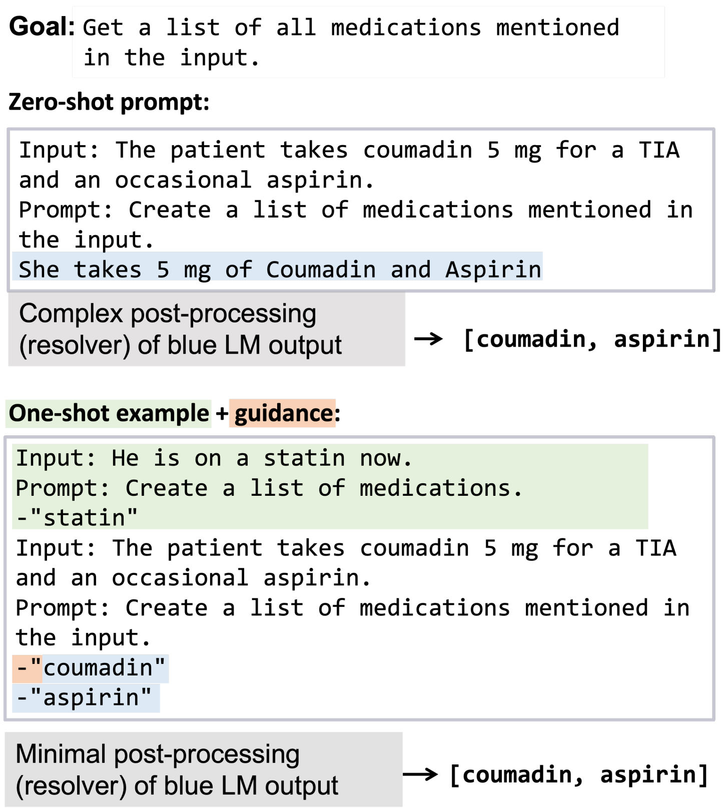
Large Language Models for Clinical Information Extraction

Acknowledgement of Co-authors I would like to acknowledge co-PhD students Stefan Hegselmann and Hunter Lang, who aided in many aspects of this chapter, particularly the creation of annotated data, the implementation of baselines, and running experiments.

7.1 Introduction

In this chapter, we explore the new paradigm of *prompt-based learning* as a workaround for the minimal amount of labeled data available for clinical NLP training algorithms. In prompt-based learning (also known as in-context learning), a pretrained language model is adapted to different tasks via priming on natural language prompts—pieces of text that are combined with an input and then fed to the language model to produce an output for that task. This paradigm has been successful for few-shot and zero-shot learning at many general-domain tasks [35, 160, 292, 233]. However, as previously described, clinical text represents a significant distribution shift from the general domain, and it was therefore unclear whether large language models (LLMs) would have sufficient exposure to generalize to the clinical domain.

In this chapter, we benchmark how large language models (LLMs) such as GPT-3



```
Get a list of all medications mentioned
in the input.
```

```
Input: The patient takes coumadin 5 mg for a TIA
and an occasional aspirin.
Prompt: Create a list of medications mentioned in
the input.
She takes 5 mg of Coumadin and Aspirin
```

```
Input: He is on a statin now.
Prompt: Create a list of medications.
-"statin"
Input: The patient takes coumadin 5 mg for a TIA
and an occasional aspirin.
Prompt: Create a list of medications mentioned in
the input.
-"coumadin"
-"aspirin"
```

Figure 7-1: Illustration of our approach using a one-shot example (green) and guidance (brown) to create a more structured LM output (blue). This significantly reduces the necessary post-processing effort of a resolver (gray).

Task	Description	Example Text	Answer	Data
Clinical sense disambiguation	Given a note and an abbreviation, expand the abbreviation (classification)	*[...] was sent to IR for thrombolysis. Post IR, ultrasound showed that [...]*	Interventional radiology	41 acronyms from 18,164 notes from CASI [176] and 8912 notes from MIMIC [1]
Biomedical evidence extraction	Given an abstract, list interventions (multi-span identification/-generation)	*[...] paliperidone extended- release tablets and [...] with risperidone [...]*	-paliperidone extended-release tablets -risperidone	187 abstracts (token-level) and 20 newly annotated abstracts (arm identification) from EBM-NLP [196]
Coreference resolution	Given a note and a pronoun, identify the antecedent (span identification)	*[...] Did develop some tremors, however. These were well managed [...]*	some tremors	105 newly annotated examples from CASI [176] with one pronoun-antecedent pair each
Medication status extraction	Given a note, extract medications and their status, e.g., active (NER + classification)	*[...] have recommended Citrucel [...] discontinue the Colace. [...]*	-Citrucel: *active* -Colace: *discontinued*	105 newly annotated examples from CASI [176] with 340 medication-status pairs
Medication attribute extraction	Given a note, extract medications and 5 attributes, e.g., dosage, reason (NER + relation extraction)	*[...] she was taking 325 mg of aspirin per day for three years for a TIA. [...]*	aspirin: {dose: 325 mg, freq: per day, duration: three years, reason: TIA}	105 newly annotated examples from CASI [176] with 313 medications and 533 attributes

Table 7.1: Overview of the five tasks studied in this chapter and the datasets that were used.

models [35, 199] perform at clinical NLP tasks. This takes the form of four contributions:

- We introduce *three new annotated datasets* for benchmarking few-shot clinical information extraction methods, as many shared clinical corpora [183, 107, 121] have data use agreements that prevent their use with LLM APIs such as OpenAI's. The datasets were generated by re-annotating the dataset from Moon et al. [176] for new tasks.

- We show that GPT-3 models performs well in clinical NLP over a set of diverse tasks (see Table 7.1), despite not being trained specifically for the domain. By replacing the complex hand-curated domain knowledge with the natural-language output of an LLM, the engineering effort required to solve a particular extraction task can be greatly reduced.
- While LLMs have been primarily evaluated at classification and generation tasks, our tasks involve a greater variety of expected output structures, such as relation extraction (see last three rows of Table 7.1). We therefore introduce *guided prompt design* to steer the LLM towards an easy-to-structure output and *resolvers* to map from the LLM outputs to the structured label space; see Figure 7-1.
- As GPT-3 models are likely not trained on significant clinical text corpora (as they are not freely available online), we analyze sources of clinical jargon and description from which the models may have learned.

7.2 Related Work

7.2.1 Prompt-Based Learning

In prompt-based learning (also known as in-context learning), a pretrained language model is adapted to different tasks via priming on natural language prompts—pieces of text that are combined with an input and then fed to the language model to produce an output for that task.

This paradigm has been successful for few-shot and zero-shot learning at many general-domain tasks [35, 160, 292, 233]. More recently, large language models such as T0 and InstructGPT have re-configured their training objectives to explicitly encourage the model to perform well at following directions provided in prompts [233, 199].

While prompt-based learning can be extended straightforwardly to classification tasks (e.g., multiple choice), more complex tasks require creativity in their implementation [173]. For example, coreference resolution is often re-framed as classification, asking which of two antecedents a pronoun refers to [233] or whether a candidate antecedent is correct [314]. This approach requires a list of antecedent candidates, which requires an additional component (e.g., a noun phrase generator) or many—potentially expensive—queries. Span classification and named entity recognition have been similarly reframed. For example, given a candidate entity X and full model access, the entity type can be predicted via an argmax over the possible types Y of the probability of statements like "X is a Y entity" [58]. Alternatively, if only a single entity is being queried for a given input, prompting can be as simple as "What is the location"[157]; however, clinical NLP often concerns itself with extraction of multiple concepts. To extract multiple spans simultaneously, Li et al. [153] and Li et al. [152] use techniques from machine reading comprehension, relying on access to the underlying model and labeled data for training the extraction layer. While InstructGPT [199] has $\sim 2\%$ or $\leq 1k$ extraction examples in its training, the LLM output is never converted to a structured form, and extraction examples are only evaluated qualitatively for improvement over other models. That is, only results for classification and generation tasks are quantified.

7.2.2 Pretrained LMs for Clinical NLP

Clinical text differs significantly from text typically utilized in general NLP, both in syntax and vocabulary [307]. As a result, the clinical NLP subcommunity often trains domain-specific models on clinical corpora following advances in language modeling from the broader NLP community. For example, clinical neural word embeddings were trained following word2vec [171, 310, 222]. More recently, following BERT, many clinical and biomedical variations swiftly followed including ClinicalBERT, SciBERT, BioBERT, and PubMedBERT [62, 13, 14, 146, 96]. However, in several applications, researchers observed the performance gains to be marginal to none over classical methods such as logistic regression [48, 137]. Additionally, previous work has so far been unable to achieve competitive results on *biomedical* NLP tasks using domain-agnostic LLMs like GPT-3 [177, 98].

7.3 Methods

7.3.1 Predicting Structured Outputs with LLMs

In this work, we assume only query access to a large language model (i.e., no gradients, no log probabilities).

Suppose we have a set of n examples $(\{x_i, a_i\})_{i=1}^{n}$, where x_i is the input text as a string, a_i is (optional) side information as a string (e.g., which acronym to disambiguate). The outputs $y_i \in \mathbb{O}$ are unobserved (i.e., to be predicted). The output space $\mathbb{O}$ is defined per task. For example, for a binary sequence labeling task, if we let $|x_i|$ be the number of tokens in x_i, $\mathbb{O}$ is $\{0,1\}^{|x_i|}$.

Prompt-based learning requires the specification of a prompt template to be applied on the input. In this work, we handcraft our prompt templates using a set of 5 validation examples per task. Let $p_j(x, a)$ be the result of filling prompt template j with inputs x and a, and further let $f(p_j(x, a)) \in \Sigma^\star$ be the string output by an LLM on input $p_j(x, a)$. The next step involves mapping the LLM generation from $\Sigma^\star$ to the structured label space $\mathbb{O}$. For example, in classification, the *verbalizer* defines a mapping between the LLM output space $\Sigma^\star$ and the discrete set of labels $\mathbb{O} = \{1, \ldots, L\}$ using a dictionary of token/label pairs [239]. However, for our structured tasks of interest, the label space $\mathbb{O}$ is more complex, and more complicated functions are needed to map to an element of $\mathbb{O}$. We define the *resolver* R as a function $R(x, a, f(p_1(x, a)))$ that maps the combined input and LLM output to the task-specific output space $\mathbb{O}$. For example, suppose the output space $\mathbb{O}$ is a *list* of strings. Then the resolver needs to turn each output $f(p_j(x, a))$ into a list (perhaps by choosing spans of text from inside of $f(p_j(x, a))$). For example, for medication extraction we might have:

$$\begin{aligned}
&x = \textit{“switched Advil for Tylenol”}, a = \textit{“N/A”},\\
&p_1(x,a) = \textit{“Note: switched Advil for Tylenol.”}\\
&\qquad\qquad \textit{Task: List medications.”}\\
&f(p_1(x,a)) = \textit{“Tylenol and Advil”}\\
&R(x,a,f(p_1(x,a))) = \texttt{[“Tylenol”,“Advil”]}
\end{aligned}$$

We refer to the output from the resolver as Resolved GPT-3, or **GPT-3 + R**, for short. Throughout, when comparing resolvers, we place in parentheses the lines of code (LOC) in the resolver, as a proxy for complexity (defined as human effort, not runtime). The required complexity of the resolver depends largely on the cleanliness of the prompt output, and by extension the prompt itself. We introduce *guided prompt design* to simplify the resolver required for complex output. As seen in Figure 7-1, this consists of (i) a one-shot example with an output in the desired structured format (which could be incorrect content-wise), and (ii) guiding the model to use the same format. Specific constructions are found in Sections 6 and 7.

7.3.2 Large Language Models Used

In this chapter, we instantiate the LLM generation function f described in the previous subsection with models available from the OpenAI API. Since OpenAI has used varying and overlapping terminology in referring to these models [1], we establish here the terminology used within this chapter. We use *GPT-3* generically throughout, as an umbrella term for a model within the GPT-3 series of large language models. *GPT-3* is therefore used as a catch-all, and not used to describe any particular instantiation.

Within the paper, we use two different instantiations of GPT-3, and at the beginning of each section, we are explicit about which specific model engine we use. For the first task, we use the engine *text-davinci-edit-001*, which we also refer to as

---

[1] `https://beta.openai.com/docs/models/gpt-3`, https://beta.openai.com/docs/model-index-for-researchers

GPT-3 edit. To our knowledge, OpenAI has not revealed the exact training scheme on which this engine is built. For all other tasks, we use the engine *text-davinci-002*, which is part of the *InstructGPT* series of models. *text-davinci-002* was trained with supervised instruction fine-tuning of a large autoregressive language model (175 billion parameters), described partially in [199]. Of note, it was not trained with the reinforcement learning paradigm described in that same paper.

7.3.3 Dataset Annotation

In the short-term, research on clinical extraction via prompting may rely on sending data to external APIs. Since data use agreements on many existing annotated clinical datasets prohibit such activity, there is a dearth of benchmarks for the community to build on. The de-identified Clinical Acronym Sense Inventory (CASI) dataset is therefore a valuable resource, as it is “publicly available to support the research of the greater NLP and biomedical and health informatics community” [176]. CASI contains snippets of clinical notes across specialties in four University of Minnesota-affiliated hospitals. While CASI was originally annotated for acronym disambiguation, we created three new annotated datasets from existing snippets of the CASI dataset. Annotation was performed by two of the authors who have background in both clinical NLP and medicine. For each task, a set of examples was jointly annotated to establish an annotation schema, each annotator then independently labeled the same set of 105 examples using PRAnCER software [150], and the two sets were then merged via joint manual adjudication.

In the following sections, we show how to build simple resolvers for five clinical NLP tasks. We find that resolvers for guided prompts are much easier to write than resolvers for un-guided prompts. The implicit structure imposed by the prompt guidance means that resolvers for a guided prompt can be less than 10 LOC. On the tasks below, we find that GPT-3 + R matches or exceeds strong few-shot, zero-shot, and even supervised baselines.

7.4 Clinical Sense Disambiguation

Overview. Clinical notes are rife with overloaded jargon and abbreviations. `Pt` can mean patient, prothrombin time, physical therapy, or posterior tibial [289, 248]. This ambiguity impacts the utility of notes for patients, clinicians, and algorithms [139, 179]. In this section, we first evaluate clinical sense disambiguation on the CASI dataset directly and then transfer a model distilled via weak supervision to another dataset.

Dataset 1. The Clinical Acronym Sense Inventory dataset consists of 500 text examples for each of 75 acronyms [176]. Due to noise in the dataset (e.g., duplications), it is common to filter to a subset of the dataset; we follow the filtering from Adams et al. [1], leading to a subset of 18,164 examples and 41 acronyms for evaluation. Similar to other works, we treat the task as multiple-choice.

Dataset 2. We additionally use a reverse substitution dataset [1] generated over the MIMIC-III Critical Care Database [121]. In *reverse substitution*, labeled data is generated from unlabeled text by replacing expansions (e.g., *physical therapy*) with their acronyms (PT) and using the original expansion as the label. We evaluate on their 8912 test examples over the same 41 acronyms as the CASI subset. Since we cannot query *GPT-3* on this dataset, we distill and transfer a model trained on the outputs from Dataset 1.

Prompting + Resolver. We used *GPT-3 edit* (using engine *text-davinci-edit-001*) with greedy decoding (temperature = 0). For each example, we provided the full clinical snippet and appended the single instruction `Expand the abbreviation:{abbr}`. Since we did *not* provide the LLM with the answer choices, the form of the output string could still differ slightly from all the candidate answers (e.g., editing "RA" to "right atria" when "right atrium" was expected). In the resolver, we choose the answer choice with the highest contiguous character overlap with the LLM generated output.

Model Distillation via Weak Supervision. Direct deployment of large language models can be difficult due to model size and data privacy. To remedy these issues, we follow several recent works [143, 256, 286] and show that we can instead view the LLM

Algorithm	**CASI Acc.**	**CASI Macro F1**	**MIMIC Accuracy**	**MIMIC Macro F1**
Random	0.31	0.23	0.32	0.28
Most Common	0.79	0.28	0.51	0.23
BERT (from Adams et al. [1])	0.42	0.23	0.40	0.33
ELMo (from Adams et al. [1])	0.55	0.38	0.58	0.53
LMC (from Adams et al. [1])	0.71	0.51	0.74	**0.69**
GPT-3 edit + R: 0-shot	0.86	0.69	*	*
GPT-3 edit + R: 0-shot + distillation	**0.90**	**0.76**	**0.78**	**0.69**

Table 7.2: **Clinical sense disambiguation.** Accuracy and macro F1 for zero-shot language modeling approaches on a subset of the Clinical Acronym Sense Inventory (CASI) data set [176] and the MIMIC Reverse substitution dataset [1]. GPT-3 is not run on MIMIC due to the data use agreement. To evaluate on MIMIC we distill GPT-3 + R into a smaller model by treating the outputs as weak supervision and following Lang et al. [144] "+ distillation", then evaluate the smaller model on MIMIC as well.

+ resolver system as a *labeler* rather than as a *classifier*, and that this can even boost performance. In particular, we use outputs of this system on CASI as weak supervision [e.g., 218] to train a smaller, task-specific model. Here we fine-tune PubMedBERT [96] and follow Lang et al. [143]; details and hyperparameters are found in the appendix.

Baselines. We compare the performance of our approach to other zero-shot language modeling methods: (i) Latent Meaning Cells (LMC), a deep latent variable model from Adams et al. [1] which is pre-trained on millions of notes from MIMIC, (ii) ELMo pre-trained on the same dataset [207], and (iii) Clinical BioBERT [13]. Numbers for these three baselines are taken from Adams et al. [1]; for all three, they choose the answer choice with the most similar representation to the contextual representation of the acronym. We also show performance for random guessing and choosing the most common answer choice per acronym (since the expansions of many acronyms follow a long-tailed distribution).

Evaluation. Accuracy and macro F1 are calculated per acronym and averaged over all acronyms (see left of Table 7.2). On CASI, GPT-3 edit + R alone already clearly outperforms the LMC model on both metrics, and the addition of weak supervision with PubMedBERT further boosts this performance. On the MIMIC Reverse Substitution dataset, despite being transferred to a new domain, our weakly-supervised PubMedBERT model performs similarly to LMC [1], which was pre-trained specifically on the MIMIC distribution. This indicates we can use GPT-3 edit + R to label a public dataset, distill its labels into a smaller task-specific model, and then transfer that model to a private dataset to obtain competitive performance. Since the CASI dataset is publicly accessible, one possible caveat is that the dataset could have been in the language model's training data; to investigate further (see Section D.2.5), we prompt the LLM on acronyms *not in the original annotations.*

7.5 Biomedical Evidence Extraction

Task Overview. Evidence-based medicine (EBM) involves synthesizing findings from across clinical research studies, but the current rapid clip of research makes it nearly impossible to keep up with all studies [231, 23]. Therefore, automated approaches for parsing clinical abstracts could aid the adoption of EBM [282, 196]. Here, we focus on extracting interventions and controls (which we will refer to just as Intervention), where the underlying goal is to identify the distinct arms of a clinical trial [196]. Token-level classification is often used as a proxy for this goal, but distilling identified spans into distinct interventions is non-trivial and often requires significant domain knowledge. Prior work on the identification of the arms of clinical studies has attempted to use coreference resolution [71] and to identify of pairs of spans with redundant information [196].

Dataset. We assess intervention identification from the angles of (i) the token classification proxy task and (ii) the underlying task of arm identification. For (i), we use the token-level annotations provided in version 2 of the dataset from Nye et al. [196] and evaluate on the 187 test abstracts provided. The average Cohen's κ was only 0.59 on this set. For (ii), the two annotators from Section 7.3.3 manually derived a list of the intervention-control arms for 20 abstracts in the test set, with perfect agreement.

Prompting + Resolvers. We use a single prompt with InstructGPT (engine *text-davinci-002*) and greedy decoding. The resolver for the token-labeling task removes noisy tokens (stop words) from the LLM output, maps remaining tokens in the output

Algorithm	**Token-level F1**	**Abstract-level Accuracy**
PubMedBERT-CRF (sup)	**0.69**	0.35
LSTM-CRF (sup)	0.65	*
GPT-3 + R: 0-shot	0.61	**0.85**

Table 7.3: **Biomedical Evidence Extraction**. Test F1 scores on the binary token-level sequence labeling problem for Intervention identification [196], and abstract-level accuracy at study arm identification. The supervised baselines were trained on 4,800 abstracts.

to the original input and labels those as 1, and merges fractured spans. The full process can be found in Appendix D.2.6. For the arm identification task, resolving simply involved splitting the output string on new lines.

Comparison. We compare to supervised approaches that train on the 4800 labeled training examples from Nye et al. [196]. PubMedBERT with an additional classification layer (LSTM or CRF) achieves close to state-of-the-art performance on the full task [96]. Since prior works report numbers combined over multiple classes, we re-run training on only the Intervention label using PubMedBERT-CRF. We also include the best supervised baseline from Nye et al. [196], an LSTM-CRF over word and character-level embeddings.

Token-level results (Proxy Task). We first evaluate sequence labeling precision at the token-level (F1 in Table 7.3). Resolved GPT-3 performs respectably compared to supervised deep baselines, but underperforms on these token-level metrics. Many error modes occur due to particularities of the schema, e.g., including extra details (like dosing schedule or route of administration) and only including an acronym or its expansion, but not both. A clarifying example can be found in Section D.2.6.

Arm Identification Results. To measure arm identification accuracy, we evaluated whether the number of arms was accurate and manually checked whether the main differentiator of each intervention arm was captured, similar to Ferracane et al. [71]. For the PubMedBERT baseline, in order to distill the identified spans to a list of arms, we assume (i) oracle splitting of spans into arms (given a span describing multiple arms, we can correctly split the span) and (ii) near-oracle coreference resolution (given multiple spans describing the same arm, we can correctly merge). Resolved GPT-3 successfully identified the correct number and content of the arms in 17 of the 20 examples. The three examples it missed were also missed by PubMedBERT. Assuming oracle splitting and coreference (a nontrivial task), PubMedBERT would still have issues with 10 further examples. Details of the evaluation and error modes are in Section D.2.6.

7.6 Coreference Resolution

Task Overview. Coreference resolution involves grouping noun phrases that refer to the same underlying entity (e.g., a person, a medical concept), and it is considered particularly important for clinically accurate information retrieval and summarization [325]. For example, when surfacing past medical history, it is critical to correctly parse pronouns to understand whether the history describes the patient or a family member.

Dataset Description. In clinical NLP, coreference resolution has been largely evaluated on the 2011 i2b2/VA challenge, which consists of thousands of coreference *chains* [278]. Due to i2b2's data use agreement, the two annotators annotated a new dataset using CASI snippets, with 5 coreference pairs for prompt design and 100 pairs for evaluation [176]. We prioritized difficult examples by focusing on pronoun coreference, where the input is a pronoun, the output its antecedent, and no tokens overlap between the two. More details are in Section D.2.2.

Prompting and Resolvers. We used the 5 examples for prompt design with InstructGPT (engine *text-davinci-002*) and greedy decoding (temperature = 0). We use a guided 1-shot prompt, where we provide an example input and begin a formatted response: "{pronoun} refers to". For 1-shot, we experiment with both correct (the true noun phrase) and incorrect answers (a random incorrect noun phrase preceding the pronoun) in the example input to tease apart the effect of the example answer versus the example formatting. To clarify that effect, we average over results from 5 different 1-shot examples. We also compare to an *unguided* zero-shot prompt, which simply appends "What does {pronoun} ... refer to?" to the input. The zero-shot

Algorithm	**Recall**	**Precision**
Toshniwal et al. [273, 274]	0.73	0.60
GPT-3 + R (50 LOC): 0-shot	**0.78**	0.58
GPT-3 + R (1 LOC): 1-shot (incorrect)	$0.76_{.02}$	$\mathbf{0.78}_{.04}$
GPT-3 + R (1 LOC): 1-shot (correct)	$0.75_{.04}$	$0.77_{.04}$

Table 7.4: **Coreference Resolution**. Macro unigram recall and unigram precision of methods on our newly annotated task using CASI [176]. The end-to-end baseline was trained on three non-clinical coreference resolution datasets and transferred to this new setting. 1-shot results are averaged over 5 prompts.

resolver involves mapping tokens back to the input due to potential paraphrases; the one-shot resolver involves only the removal of a single quotation mark, making the guided prompt easier to resolve. Section D.1.3 contains more detail on the prompts.

Comparison. We compare to deep end-to-end coreference resolution, as it has been shown to perform well [148]. In particular, we compare to the *longdoc* model from [273], which trained on multiple coreference datasets in order to generalize to new settings.

Results.

We evaluated via macro unigram recall (% of label's unigrams in the resolved output) and unigram precision (% of unigrams in the resolved output in the label) (Table 7.4). We tokenized using Stanza [213] for these metrics. While the *longdoc* baseline trained on thousands of non-clinical coreference examples performed considerably well already, it is outperformed by Resolved GPT-3. We found the 1-shot example mostly constrains the LLM output to quoting (rather than paraphrasing); without guidance, the LLM may output e.g., "The antecedent is unclear." Further, the accuracy of the 1-shot example was irrelevant to the performance, an observation previously reported in the classification setting, now seen for span extraction [172].

7.7 Medication Extraction

The recognition of clinical concept mentions (problems, treatments, etc.), their modifiers (e.g., negation), and relations (e.g., dosage) is a fundamental building block in

Algorithm	**Recall**	**Precision**
ScispaCy [189]	0.73	0.67
GPT-3 + R (32 LOC) (0-Shot)	0.87	0.83
GPT-3 + R (8 LOC) (1-Shot)	$\mathbf{0.90}_{.01}$	$\mathbf{0.92}_{.01}$

Table 7.5: **Medication extraction.** Micro recall and precision for medication extraction on our self-annotated dataset.

Algorithm	**Conditional Accuracy**	**Conditional Macro F1**
T-Few (20-shot)	0.86	0.57
GPT-3 + R (32 LOC) (0-Shot)	0.85	0.69
GPT-3 + R (8 LOC) (1-shot)	$\mathbf{0.89}_{.01}$	$0.62_{.04}$
GPT-3 + R (8 LOC) (1-shot) + added classes	$0.88_{.02}$	$\mathbf{0.71}_{.03}$
GPT-3 + R (8 LOC) (1-shot) with shuffled classes	$0.88_{.01}$	$0.66_{.03}$

Table 7.6: **Medication status classification.** Conditional accuracy and macro F1-score for Identification of medication status *active*, *discontinued*, and *neither*.

clinical NLP [119]. Here we examine the extraction of medication concepts with two different schemas.

7.7.1 Recognition + Status Classification

Here we extract a list of medications and label each with a status modifier: active, discontinued, or neither (e.g., allergy, proposed medication).

Dataset description. We created new annotations for medication and status on top of CASI Moon et al. [176]. The examples were enriched for changeover in treatment. For 105 randomly selected snippets, the annotators extracted all medications mentioned and classified its status with one of the 3 labels. Further details are in Appendix D.2.3. Unlike in Section 7.7.2, all mentions corresponding to the same medication are collapsed.

Prompting and Resolver. We again used 5 examples for prompt design with InstructGPT (engine *text-davinci-002*) and greedy decoding. Our prompt asked the

Subtask	Algorithm	Medication	Dosage	Route	Frequency	Reason	Duration
Token-level	PubMedBERT + CRF (Sup.)	0.82	0.92	0.77	0.76	0.35	**0.57**
	GPT-3 + R: 1-shot	**0.85**	0.92	**0.87**	**0.91**	**0.38**	0.52
Phrase-level	PubMedBERT + CRF (Sup.)	0.73	0.78	0.71	0.41	**0.22**	**0.30**
	GPT-3 + R: 1-shot	**0.75**	**0.82**	**0.81**	**0.87**	0.21	0.25
Relation Extraction	PubMedBERT + CRF + Shi and Lin [247] (Sup.)	*	0.67	**0.65**	0.36	0.19	**0.21**
	GPT-3 + R: 1-shot	*	**0.80**	0.63	**0.60**	**0.34**	0.16

Table 7.7: **Medication attribute extraction.** F1 scores on our newly annotated medication extraction dataset. The baselines are trained using supervised learning on i2b2 [276], then transferred to the test domain. *Relation Extraction* additionally requires the model to match modifiers (dosage, route, etc.) to the medication span. Baseline end-to-end relation extraction performance suffers due to errors cascading from the extraction step.

model to simultaneously output the list of medications and the status of each. We evaluate the prompt in an unguided zero-shot manner and in a guided one-shot manner. Further, to clarify the effect of the 1-shot example on modifier accuracy, we examine how status classification performance changes if we (i) artificially augment the 1-shot example so all three status classes are observed, and (ii) whether the statuses need to be correct, or just present. We averaged over 5 different 1-shot inputs to clarify these effects; each 1-shot example contained between 3 and 8 medications. We describe the resolvers for the zero- and one-shot cases in detail in Section D.2.8; the former involved several regular expressions, and the latter required only a few short lines.

Comparison. We used a rule-based method as a medication extraction baseline, since historically they perform well [257]. To this end, we leveraged the Python library ScispaCy with the `en_core_sci_sm` package for entity recognition [189, details in Appendix D.2.8].[2] For medication status classification, we compare to `T-Few` [159], a few shot LLM method fine-tuned on a set of additional snippets we labeled from the same distribution (20 snippets containing 60 medication statuses). This method predicts the status, *given the token indices for each medication.*

Results. Table 7.5 shows micro recall and precision for medication extraction; we count a prediction as correct if the predicted string exactly matches one. Overall,

---

[2] We do not use a supervised baseline trained on the i2b2 2009 challenge data (as in Section 7.7.2) because their schema purposefully excluded medications in the *Neither* category.

Resolved GPT-3 outperforms the ScispaCy linkage baseline consistently by a considerable margin. The addition of the 1-shot example greatly improves precision, since in the 0-shot case, some GPT-3 outputs included extraneous extractions (e.g., a procedure). Typical failure modes of the baseline include incorrect recognition of overloaded abbreviations and missing vendor-specific drug names. Table 7.6 shows *conditional* accuracy on medication status classification. For an apples-to-apples comparison, we conditioned on the subset of medications found by all GPT-3 methods (241/340) and evaluated T-few on that subset as well. We find that if the rarer *Neither* class wasn't demonstrated in the 1-shot example, it was unlikely to be output, depressing the F1 score; including all classes in the 1-shot prompt appears more important than necessarily having the correct labels.

7.7.2 Recognition + Relation Extraction

Dataset description. The annotators created a second new dataset for medication extraction from the snippets from Moon et al. [176]. The annotators closely followed the schema from the 2009 i2b2 medication challenge [276], with small deviations explained in Appendix D.2.4. For 105 randomly selected snippets, the annotators labeled mentions of medications, dosages, routes, frequencies, reasons, and durations, if available, and their correspondences. We examine the task from three different framings: a token-level annotation task, a phrase-level annotation task, and end-to-end relation extraction. Let's consider the example phrase "takes, Tylenol twice daily". For the token-level task, the desired output is a classification for each of the 4 present tokens, namely *[None, Med, Frequency, Frequency]*. For the phrase-level task, our goal is to annotate not just each token, but group together tokens that belong to the same phrase. To do so, we use the BIO (Beginning, Inside, Ouside) schema; here, the desired output would be *[O, B-Med, B-Frequency, I-Frequency]*, since "twice daily" should be treated as a single entity. Finally, for the end-to-end relation task, we want to not only identify medications and their modifiers, but match each modifier to the corresponding medication. Therefore, the desired output is *Medication: "Tylenol", Frequency: "twice daily".*

Prompting and Resolver. We again used 5 examples for prompt design with InstructGPT (engine *text-davinci-002*) and greedy decoding (temperature = 0). We use a different guided 1-shot prompt (containing 7 entities each) for each of the three framings outlined above; these can be found in Appendix D.1. The resolvers for all were short.

Comparison. For token and phrase-level classification, we used a PubMedBERT model topped with a CRF layer. For end-to-end relation extraction, we first used the token-level baseline to extract entity spans, then used the technique from Shi and Lin [247] to classify whether each pair of entities was related. We then postprocessed these pairwise outputs to match modifiers to their medications. For all the three tasks, since we followed the 2009 i2b2 medication extraction annotation guidelines, we fine-tuned the baselines with labeled data from i2b2 (10 fully annotated notes with 154 medication mentions, which we postprocess into smaller annotated chunks) and directly evaluated them on our datasets. [276]. Appendix D.2.9 contains more detail for the baselines and evaluation.

Results. Table 7.7 shows that the 1-shot GPT-3+R outperforms the i2b2-supervised baseline across all task framings. The baseline end-to-end relation extraction performance suffers due to cascading extraction errors, as the longest token in the medication name had to be matched. GPT-3+R struggles with the *duration* and *reason* entities; however, it has been previously found that there is often large disagreement (F1 estimated 0.2–0.5) in inter-annotator agreement for these two entities, since they tend to be longer with ambiguous boundaries.

7.8 Analysis of Common Crawl Data

The experiments in the preceding sections indicate an ability of the GPT-3 series of models to understand and parse clinical note text. However, given that there is a lack of significant clinical note corpora available publicly online, this begs the question of what training data these models used in order to learn clinical jargon. In this section, we characterize the constituents of the data used to parse clinical contexts.

Data

The original GPT-3 language model (that the instruct-based models were built on top of) was trained autoregressively on a scrape of Common Crawl filtered for quality and duplication (60% of training data) and curated corpora (e.g., Wikipedia) for the other 40% [35]. As we do not have access to this dataset, we instead turn to the 750GB C4 dataset as a proxy [214]. Compiled for training the T5 model, C4 stands for ”Colossal Clean Crawled Corpus.” Like the GPT-3 dataset, C4 underwent filtering to restrict to de-duplicated, higher-quality, non-offensive English language text. While C4 is significantly smaller than the corpus used to train GPT-3, we believe it can provide insights into the sources that constitute health data in the Common Crawl.

We leverage the C4 search provided by The Allen Institute for Artificial Intelligence[3]. Given a set of query terms, the search provides up to 10,000 results from the C4 dataset; they do not specify how these results are ordered. To understand what sources of clinical jargon might exist and how LLMs might have learned clinical associations, we focus on extracting documents in which clinical short forms (e.g., “dc”, “pt”) occur *alongside* their expanded long forms (e.g., “discontinue”, “prothrombin time”). The clinical short forms alone are insufficient, as e.g., “dc” also commonly refers to the city, the comic publisher, and the clothing brand. If searching for “dc”, only a tiny fraction of the 10,000 might use the clinical context. On the flip side, only using the long form may skew towards more formal settings (e.g. research articles and medical reference materials) that do not transfer to the clinical text setting.

Therefore, the first step was compilation of a set of pairs of short forms and corresponding long forms. 92 of these were taken from the filtered disambiguation dataset described in Section 7.4. We excluded (i) those acronyms that included typos in the short form (e.g., used “AB” instead of “AV”), (ii) non-clinical long forms (e.g. “masters” for MS, or “United States” for US), or (iii) long forms that didn’t appear at least once. While non-clinical long forms are important for the disambiguation task, they do not shed insight on the distribution of clinical text. In addition, we added ten common additional shortenings that are more clinically colloquial in nature

---

[3] `https://c4-search.apps.allenai.org/`

and do not necessarily correspond to formal shortenings (e.g., "PT" is a formally used shortening for "physical therapist"). These ten additional shortenings were: "carbo (carboplatin)", "vanco (vancomycin)", "qid (four times daily)", "qhs (at bedtime)", "prn (as needed)", "cc (chief complaint)", "ddx (differential diagnosis)", "dx (diagnosis)", "fx (fracture)", and "sq (subcuteaneous)." We refer to this subset as *Colloquial* in results.

For each pair, we scraped both the number of total search results and the contents (URL + text snippet) of the first 100 results. After scraping, we de-duplicated entries, on the basis of URL. Of particular note are two entries which had no search results. One was the drug administration route "per rectum"; this can be fully explained by the C4 filtering of potentially offensive terms, of which "rectum" is one. The other was "retrograde tachycardia," which a Google search reveals rarely occurs online (164 results). The post-processing resulted in a total of 8,611 scraped results over 100 pairs, if we exclude the two pairs with no results.

Characterization of Sources

Next, we characterized the sources of the scraped results. To do so, we manually examined and classified approximately 60 results from a variety of pairs to emerge with 10 categories: `Patient Forum`, `Personal Blog`, `Research Article`, `News Article`, `Clinician Forum`, `Patient Health Resource`, `Medical Encyclopedia/Dictionary`, `Commercial Health`, `Non-Health`, and `Other Health.` Existing website classification software (e.g., Google Cloud Natural Language API) did not provide sufficient specificity to fully distinguish these categories, and therefore, we turned to a GPT-3 series model, `text-davinci-003` for zero-shot URL+ snippet classification. We used the prompt below, restricting text snippets to their first 3000 characters to control costs.

> *Classify the URL and text below into one of the following categories: Patient Forum, Personal Blog, Research Article, News Article, Clinician Forum, Patient Health Resource, Medical Encyclopedia/Dictionary, Commercial Health, Non-Health, or Other Health.*
>
> *URL:* <Insert URL>

PT, physical therapy	10,000+
RA, rheumatoid arthritis	10,000+
PT, prothrombin time	2347
fx, fracture	1231
LE, lymphedema	667
RA, room air	457

PT, posterior tibial	315
CC, chief complaint	281
ddx, differential diagnosis	139
LE, leukocyte esterase	63
NAD, nothing abnormal detected	14
PR, per rectum	0

Table 7.8: Number of search results for a subset of the clinical text pairs, as ascertained from the AI2 utility for searching Colossal Clean Crawled Corpus (C4).

Text: <Insert Text>

Classification:

Of the 8611 queries, completions from 8605 queries fell automatically into one of the ten stated categories. Three inputs resulted in error outputs (which we excluded from further analysis), two outputted "Clinical Forum" which we mapped to `Clinician Forum`, and one outputted "Clinical Trial" which we mapped to `Research Article`. A spot check of ten examples spanning the different classes, the GPT-3 completions proved accurate. Further, we look at tokens within the URL, and find that the distributions of terms across classes generally match expectation; details can be found in Appendix D.4.

Results

Table 7.8 lists the number of search results for 12 of the clinical text pairs queried, spanning a range from 0 to over 10,000. The scarcity of certain terms cements the need to understand whether the test time data is adequately reflected in the training data distribution.

Table 7.9 showcases the distribution of sources of clinical text pairs, split between the CASI dataset (mostly acronyms, initialisms) and the common *Colloquial* set of pairs described previously. While `Research Articles` form the plurality of sources for both sets of pairs, it is interesting to note that they are a far less common data source for the colloquial terms (median of 34% of mentions for CASI, 16% for *Colloquial*). Going one step further, there are several colloquial terms that are found even more rarely in research articles—only 8% of *fx (fracture)* mentions and only 7% of *ddx (differential diagnosis)* mentions. In contrast, texts from `Clinician Forums` become much more valuable for this *Colloquial* data (median of 2% for CASI, 13% for *Colloquial*). The final column indicates that ultimately, the distribution of source text differs widely depending on the jargon itself.

Takeaways

Our analysis of the frequency of clinical mentions in the C4 dataset highlights a disparity in mentions for different clinical conditions. For example, due to the content filtering mechanisms in place, many genitourinary conditions would not be well represented in the C4 data. Specifically, this indicates a need for more precise filtering mechanisms to ensure a broad base of knowledge. More broadly, this indicates a need for practitioners to probe what corpora comprise the datasets on which foundation models are built, as upstream decisions can have large effects.

Our analysis of the distribution of health text sources indicate that a wide variety of resources (e.g., research articles, patient and clinician forums, patient health resources) are required in order to ensure proper coverage of clinical terminology, particularly for the kind of colloquial language found in clinical text. Given the current security concerns of sharing LLMs trained on patient notes, the findings in this section could serve as a blueprint for sources of training data for open-sourced clinical language models.

7.9 Conclusion

In this chapter, we introduced new annotated datasets to show that (i) large language models have great promise at diverse clinical extraction tasks and (ii) we can guide generations to map to complex output spaces with only light post-processing. We also demonstrated how weak supervision over the system's outputs can be used to train smaller, task-specific models that are more deployable. Finally, we explored what data sources in Common Crawl GPT-3 might be learning clinical jargon from.

	CASI Median	**Colloquial Median**	**Maximum Observed***
Research Article	34%	16%	89% (LA, left atrial)
Patient Health Resource	20%	15%	92% (MS, multiple sclerosis)
Commercial Health	9%	14%	59% (US, ultrasound)
Medical Encyclopedia/ Dictionary	4%	5%	60% (SA, sinuatrial)
Clinician Forum	2%	13%	37% (dx, diagnosis)
Personal Blog	1%	4%	24% (AMA, advanced maternal age)
News Article	2%	4%	16% (LE, lymphedema)
Patient Forum	0%	3%	37% (carbo, carboplatin)
Other Health	0%	1%	8% (DC, discharge)
Non-Health	3%	1%	69% (OP, operative)

Table 7.9: Distribution of the sources of data in which clinical text pairs are found, as classified by `text-davinci-003`. *CASI median* refers to a set of 100 pairs derived from the CASI dataset, and *Colloquial Median* refers to a set of 10 pairs of common, manually created clinical text jargon. *For maximum observed, we exclude query pairs with fewer than 15 results, to prevent signal from a noisy handful of examples.

The scope of clinical NLP extends past what we studied here, and important next steps involve experimenting with LLMs such as OPT [321] for which we can run inference locally, enabling evaluation on existing benchmarks and fine-tuning. Another important direction involves leveraging the outputs from several prompts (e.g., 1-shot prompts with different examples) to learn to determine when GPT-3 is uncertain; this increased reliability will be vital given the high-stakes in clinical information extraction. Taken as a whole, our work indicates a new paradigm for clinical information extraction—one that can scale to the lofty goals of clinical NLP.

7.9.1 Limitations

While large language models show great promise at clinical information extraction, there are clear limitations to their use. First, it is still difficult to guide a LLM to match an exact schema—clinical annotation guidelines are often multiple pages. We found that even when the Resolved GPT-3 outputs were impressive qualitatively, they did not always match at the token-level. For example, in tagging durations, one Resolved GPT-3 output was "X weeks" instead of "for X weeks". While this particular omission is trivial, it highlights the difficulty of communicating nuanced guidelines. The reliance on prompt design is both a limitation and an area for future automation.

Second, we found a bias in GPT-3 towards outputting a non-trivial answer even where none exists. For example, for medication extraction the prompt we ended up using was, "Create a bulleted list of which medications are mentioned and whether they are active, discontinued, or neither." However, prior to this we had experimented with two separate prompts: "Create a bulleted list of *active* medications, if any." and "Create a bulleted list of *discontinued* medications, if any." If there was one active and one discontinued medication, the respective LLM outputs would be correct. However, if there were two active medications and none discontinued, the LLM primed with the discontinuation prompt tended to try to find an output and usually resorted to listing one or more active medications. Therefore, it is important to craft prompts or tasks that avoid this pitfall. For example, this could be achieved via (i) chaining multiple prompts, e.g., first asking if a certain entity type exists in the input, before asking for a list [153, 308] or (ii) using an output structure like the sequence tagging approach.

Finally, because of the data use restrictions on most existing clinical datasets, which prohibit publicly sharing the data (e.g., to the GPT-3 APIs), all tasks except for biomedical evidence extraction were derived from the publicly-available CASI dataset [176]. While we show the promise of transferring to a new setting in Section 4, it would be ideal to have been able to directly evaluate on multiple hospital systems at multiple points throughout time. Clinical text in CASI was drawn from notes from several hospitals and a diverse set of specialties, but is by no means representative of all clinical text. For example, the CASI paper states that the notes were "primarily verbally dictated and transcribed," but this practice is not universal. Further, as is unfortunately common in clinical NLP, we only tested in English, leaving testing the ability of LLMs to operate in different languages to future work [191].

7.9.2 Ethical Considerations

The datasets introduced in this paper involved only new annotations on top of existing, publicly available clinical text. Dataset annotation was conducted by two authors of the original work described in this chapter, and therefore there are no associated concerns, e.g., regarding compensation. As discussed in limitations, we believe these new annotated datasets serve as a starting point for the evaluation of LLMs on clinical text, but we concede that conclusions about specific performance cannot be ported to other languages, hospital systems, or temporal settings (as clinical text is quite subject to dataset shift).

If large language models were to be integrated into clinical extraction pipelines, as presented in this paper, there are large potential benefits. Clinical text is being created at a scale far too large for manual annotation, and as a result, cohorts for clinical study are largely small and hand-curated. Automatic structuring of clinical variables would help catalyze research that may be prohibitively expensive otherwise – allowing for study of rarer or less funded diseases as well as the analysis of real-world evidence for subpopulations that may not be observed in clinical trials. However, due to the high-stakes setting, it is imperative that the performance of such a system is evaluated in the same environment it will be used in, and that the performance numbers are stratified by cohorts of note (e.g., racial, socioeconomic, patient comorbidities, disease stage, site of care, author's clinical role and seniority); such variables were not available in the data we used here.

In this chapter, we accessed the GPT-3 model using the OpenAI API alone. However, we acknowledge that even the inference cost is still nontrivial (see Appendix D.3). We presented in Section 4 a paradigm of using weak supervision to distill a much smaller model, using pseudolabels learned from GPT-3, and we encourage such work to mitigate the environmental impact of deployment.

We also want to highlight the inherent dangers of using large language models in higher stakes settings than clinical information extraction. As misinformation (particularly autogenerated misinformation) is on the rise, these models will be increasingly trained on untrue material. Even a question as benign as "What treatments were taken for the patient's COVID-19" could yield a complicated response.

Finally, given the increased ease of clinical information extraction, we caution against hasty data science that forgoes robust evaluation. While these models are impressive in the few-shot setting, it is important to conduct thorough error analysis to understand pitfalls and biases.

Chapter 8

Introducing Autocomplete for Cleaner Electronic Health Records

Acknowledgement of Co-authors I am the second author on this paper, following first author and my Master's supervisee Divya Gopinath. While Divya and I had near-daily conversations about the work described in this chapter, she contributed almost the whole of the software development as well as the bulk of the writing.

8.1 Introduction

Clinicians currently spend more time documenting information in electronic health records (EHRs) than communicating with patients, and the timesink in using inefficient EHRs is posited to be a leading cause of physician stress and burnout [41, 82]. Doctors prefer using natural language and free-text for documentation over restrictive structured forms [128], but clinicians have adapted to time-intensive note-writing by relying on overloaded acronyms and jargon [254].

Consequently, medical documentation is often noisy, ambiguous, and incomplete. The lack of structure in notes further hinders understandability for patients, other physicians, and machines [11, 84, 133]. The information within EHR notes remains largely untapped and, at present, cannot be easily used for downstream medical care or for machine learning models that rely on structured data.

8.1.1 Contributions

We propose a method called *contextual autocomplete*, which quickly captures clinical concepts at the point-of-care via learned suggestions. To do so, we build a hierarchical language model for clinical concepts that can operate in the noisy domain of ED notes. Our model is designed to be deployed in a live hospital environment, with inputs constrained to the triage information and past medical notes available to a doctor *before* a note is written. While these constraints make it infeasible to build a generative language model, we generalize to the task of *autocompletion*, where we make multiple suggestions for the next clinical concept to document and allow the clinician to determine the correct choice. As all suggestions are mapped to standardized clinical vocabularies, we can simultaneously impose structure on notes as they are being written, disambiguate between concepts, and make documentation faster for clinicians in real-world hospital settings.

We present contextual autocompletion as the cornerstone of an intelligent EHR in Figure 8-1. The contextual autocompletion tool reduces the amount of text a clinician has to type by suggesting relevant terms using a learned context. Tagged terms uniquely identify clinical concepts and can be linked with relevant information from the medical record. Moreover, these terms can facilitate widespread improvements in documentation and reduce overall cognitive load on doctors. Once a term is tagged, it can be automatically inserted in multiple locations within the clinical note to limit the amount of redundant information a clinician types. For example, a tagged condition in an earlier part of a note can be automatically appended to the Past Medical History section that appears later on, as in the right panel of Figure 8-1. This mitigates the “death by a thousand clicks" phenomenon that EHRs suffer from [242]. In addition, live-tagging clinical concepts can provide immediate rewards to physicians in the form of decision support (as described in the following chapter); the captured structured data can then be used to build smarter EHR interfaces that enable contextual information retrieval about disease history and lab trends, without ever leaving the note interface. Finally, tagging clinical concepts with our

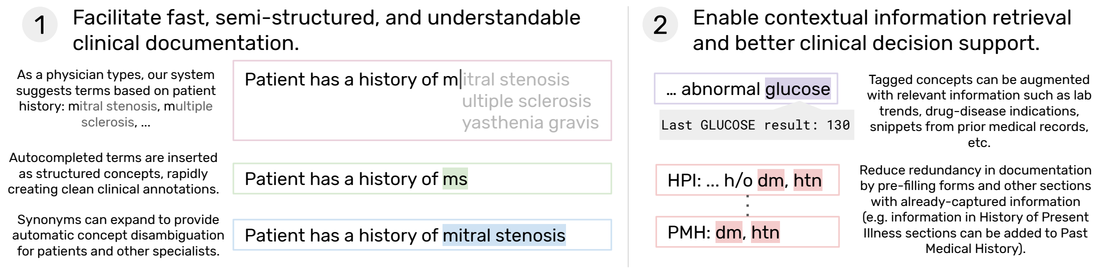

1
Facilitate fast, semi-structured, and understandable clinical documentation.
As a physician types, our system suggests terms based on patient history: mitral stenosis, multiple sclerosis, ...
Patient has a history of m|itral stenosis
ultiple sclerosis
yasthenia gravis
Autocompleted terms are inserted as structured concepts, rapidly creating clean clinical annotations.
Patient has a history of ms
Synonyms can expand to provide automatic concept disambiguation for patients and other specialists.
Patient has a history of mitral stenosis
2
Enable contextual information retrieval and better clinical decision support.
... abnormal glucose
Last GLUCOSE result: 130
Tagged concepts can be augmented with relevant information such as lab trends, drug-disease indications, snippets from prior medical records, etc.
HPI: ... h/o dm, htn
PMH: dm, htn
Reduce redundancy in documentation by pre-filling forms and other sections with already-captured information (e.g. information in History of Present Illness sections can be added to Past Medical History).

Figure 8-1: Semi-structuring notes with contextual autocompletion can enable extensive changes to both documentation and clinical decision support.

system allows for the translation of acronyms and domain-specific language to common names. By normalizing key clinical concepts from notes to a universal vocabulary (the Unified Medical Language System, or UMLS), notes written using our system are semi-structured and parseable. Building a tool that allows clinicians to document terms on-the-fly not only decreases documentation burden, but also curates large-scale prospective datasets of labeled clinical concepts (e.g., conditions, symptoms, labs, and medications) in notes.

8.2 Background: The Current State of Clinical Workflows

When a patient enters the Emergency Department (ED), there are several phases of documentation. First, a triage nurse records patient vitals and a short description of the visit reason. This triage note is then summarized in a succinct phrase known as the Chief Complaint. Doctors also maintain a clinical note which is updated throughout the course of the visit and contains information about the patient's history, current presentation, pertinent labs and tests, and a final diagnosis and treatment plan. This note is also a constantly-evolving document. It is edited before the doctor sees the patient (to document patient history), while treating a patient (to document relevant symptoms and tests), and after the patient is discharged (to document the final diagnosis). The note is time-intensive to create, and as such, our work focuses on decreasing documentation burden within the doctor's note.

Clinical staff also have access to the patient's past EHR, which is a rich data source. The bulk of information in EHRs lies within unstructured clinical notes in the patient's file, which contain detailed information about disease history and prior clinical care. Yet these notes are long and difficult to quickly parse. In our dataset, the median number of EHR notes per person is 34 with a median note length of 301 words. There have been attempts to mitigate this information overload by creating semi-structured representations of a patient's medical history such as the problem list, which catalogs a patient's prior conditions. However, these lists are poorly maintained and inconsistent amongst practitioners [280].

Efforts to intelligently structure free-text within clinical notes have been limited. One common technique is to pre-fill notes with templates that rely on structured text [294]. For example, clinical notes usually begin with a summary of patient demographics and the chief complaint like `26 y/o M complains of dyspnea`. This method works for routine cases and structured, repetitive phrases that occur in some sections of notes, but fails to capture subtleties of documentation that reflect the nuances of clinical reasoning and physician preference.

To date, the largest-scale attempt to ease clinical documentation burden with machine learning is by Greenbaum et al. [94], who built an autocomplete model to predict candidate chief complaints in the ED from a set of approximately 200 standardized options. The model—a multiclass SVM trained on triage information—was used to structure 99% of chief complaints in a live setting. We build on the work in Greenbaum et al. to provide contextual autocompletion functionality for an unstructured clinical note by architecting a model that incorporates both contemporaneous clinical information (triage text, vital signs, and laboratory results) and past medical history (EHR), and by building an interface that supports intuitive and on-the-fly documentation of multiple tagged terms from a large set of clinical concepts.

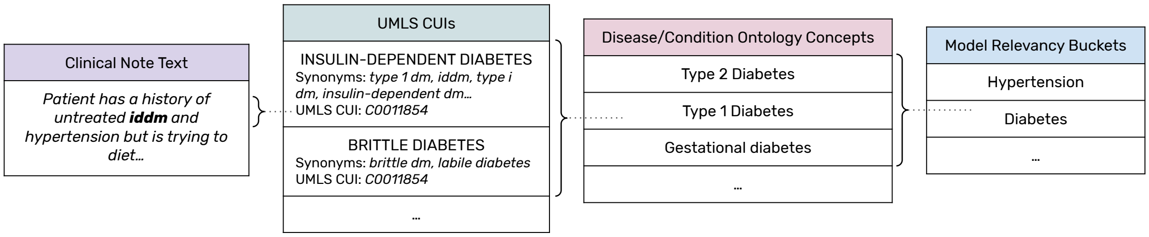

Clinical Note Text
Patient has a history of untreated iddm and hypertension but is trying to diet...
UMLS CUIs
INSULIN-DEPENDENT DIABETES
Synonyms: type 1 dm, iddm, type i dm, insulin-dependent dm...
UMLS CUI: C0011854
BRITTLE DIABETES
Synonyms: brittle dm, labile diabetes
UMLS CUI: C0011854
...
Disease/Condition Ontology Concepts
Type 2 Diabetes
Type 1 Diabetes
Gestational diabetes
...
Model Relevancy Buckets
Hypertension
Diabetes
...

Figure 8-2: Data model for our ontology of conditions. Clinical notes are normalized to UMLS, and sets of UMLS IDs (CUIs) are aggregated to create unique concepts in our ontology. Ontology entries are then grouped together in coarser model relevancy buckets.

8.3 Methods

8.3.1 Data Overview and Cohort Definitions

We use data from 273,000 anonymized visits to the Beth Israel Deaconess Medical Center (BIDMC) ED over the last decade, representing around 140,000 unique patients. For each visit, we have access to patient demographics, triage information (triage assessment, vital signs, and chief complaint), clinical notes from both the doctor and nurse assigned to the patient, and medications currently prescribed to the patient. In addition, we have all prior EHR notes for patients who had previously been in the BIDMC system (74% of visits). We do not restrict our analysis to any particular subset of these visits.

8.3.2 Defining Labels for Autocompletion

The goal of autocompletion is to predict terms that the doctor would type into a note given a clinical context. In order to create positive labels for this task, we must extract documented clinical concepts from medical notes through named entity recognition (NER) on the text. We then normalize these concepts to UMLS. An example of this is shown in Figure 8-2.

First, we restrict ourselves to a subset of the UMLS ontology and exclude terms that do not correspond to a concrete clinical concept (e.g., `Health Care Activity`). The filtered terms are then inserted into a trie data structure, which we use to identify all UMLS concepts in time linear in the note length. We also apply a modified NegEx-style negation detection algorithm as in Chapman et al. [45] to identify and mark which of these extracted terms occurred within a negative context. We provide additional details on our implementation of negative context detection in Appendix E.1.2. After filtering out concepts that appear fewer than 50 times, we extract 8,678 remaining UMLS concepts from visit notes. We then group concepts into two categories: conditions that a patient might have a history of, and symptoms that occur in the present medical context. Ambiguous acronyms such as `MS` are resolved as follows: if the term is almost always used to represent a singular concept within the ED, we default to that CUI, and otherwise ignore it. Two clinicians then independently verified these lists.

Concept disambiguation between closely linked conditions is difficult. As an example, `hyperlipidemia` and `increased LDL` are distinct UMLS concepts that encode similar semantic meanings, whose differences are not clinically meaningful in the ED and may be unknowable to the physician. To improve sample efficiency, we share weights between similar conditions by introducing a manually-curated hierarchy of UMLS terms and rolling terms up to an appropriate level of specificity such that every combined term carried the same medical meaning. A subset of this ontology for conditions is shown in Figure 8-2. The complete revised ontology (consisting of 940 entries encompassing 8,451 unique UMLS CUIs) can be found in Section E.2.4 of the Appendix.

Condition concepts also represent varying levels of granularity, which is necessary in clinical text [266]– a doctor could use `depression`, `severe depression`, or `chronic depression` to describe a patient, but these are distinct entries in our ontology. Choosing between similar terms during documentation is currently a subjective practice that depends on the clinical scenario and user-specific preference. We address this by further rolling up our ontology into a coarser set of *model relevancy buckets* which group terms corresponding to similar underlying medical concepts. We build our models to have predictive power at the level of relevance buckets, and later rank individual terms within a model relevancy bucket to suggest terms for a doctor to document. This injects a medical inductive bias that forces parameter sharing between similar concepts, thereby allowing us to leverage closely-related groups of rare conditions to learn a common predictor. A subset of the 227 model relevancy buckets can be seen in Figure 8-2. We find our UMLS-based extraction is equally effective on our text as out-of-the-box learned extraction models such as cTakes [238], scispaCy [190], and DistilBERT [232], while remaining sufficiently fast for live deployment. We elaborate on this in Section E.1.3 of the Appendix.

8.3.3 Developing Predictive Features for Autocompletion

In a typical language model, one attempts to predict the distribution $p(w_i|c_i)$ of an unknown word w_i using a *context* c_i which captures the semantic information necessary to make such a prediction. For a generative model, c_i usually consists of a complex representation of $w_{1:i-1}$ (the words preceding w_i) and is often parameterized by a deep neural network. These representations are state-of-the-art for clinical language modelling [114, 306, 161]. In our framework, complex inference techniques are too slow to surface live suggestions with low latency in a hospital setting, and we only seek to predict clinical concepts rather than the general language a clinician types.

All features we use as part of our context must be available *before* a patient and physician interact, as this allows us to surface live suggestions as the clinician creates documentation. This limits the data we can incorporate into our autocompletion models to unstructured textual data from the EHR, which is our glimpse into the patient's medical history; and triage-time information such as vitals, chief complaint, and the triage assessment.

Featurizing Textual Data

Our greatest sources of knowledge about the patient prior to clinician interaction lies in prior EHR notes and the triage assessment. To featurize prior EHR documents, we run the NER and hierarchical roll-up algorithms from Section 8.3.2. The result of this is a mapping from a clinical text T to a set of UMLS-mapped clinical concepts mentioned in the text, as well as a coarser representation of the types of conditions incorporated into the note. To encode triage assessments, we simply use a standard term frequency-inverse document frequency (TF-IDF) encoder to capture a normalized bag-of-words representation of the text.

Featurizing Triage Vitals

Triage vitals are already structured as they represent information that is inherently quantitative, such as heart rate and blood pressure. We discuss specific strategies of further preprocessing triage vitals with each model use-case below.

8.3.4 Autocompletion Models

We frame contextual autocompletion as a hierarchical, human-in-the-loop language model that suggests clinical concepts to document as a physician is typing. We leverage four pieces of data to form our context $c_i = [w_{1:i-1}, T, \mathcal{H}, V]$; namely, the text so far, the triage assessment, past EHR notes, and the patient's triage vitals. Calling an inference step of our model each time a word is written or removed is prohibitive in terms of latency, so we employ a rules-based approach to incorporate $w_{1:i-1}$ into our prediction while learning how to use the nuanced information in $T, \mathcal{H}, V$. Inference thus only needs to be run once per patient. We first use $w_{1:i-1}$ to determine when the clinician wants to enter a potential clinical concept, and if so, whether that concept is a condition, symptom, lab, or medication. We then generate four term-wise rankings for each concept type, and stack the suggested rankings for each of the concept types to generate a total ranking. In practice, we filter these rankings to entries with any synonyms that match the typed query a doctor has entered. The doctor can either continue to type or select a term, which is then inserted into the note as a tagged concept using the synonym that he/she intended– as an example, typing `ht` might give `hypertension` as a suggestion because of its synonym `htn`, and if a doctor chooses to autocomplete, we insert `htn` to preserve intended note vocabulary. We outline our four concept-specific ranking models:

1. Conditions: we learn a mapping from the triage text and the clinical concepts mined from the EHR to a ranked list of relevant prior conditions that the doctor might want to document. This autocompletion model is primarily used to write the History of Present Illness (HPI) sections of notes, where physicians note past medical history that is relevant to the current patient presentation. We find that vitals have little to no predictive power in this model.
2. Symptoms: we learn a mapping from the triage text, chief complaint, and vitals to a ranked list of relevant symptoms that the patient currently presents with. We do not include information from the patient's past medical record in our predictions because a patient's current presentation is only loosely related to prior visits.
3. Labs: we simply list labs by their recorded frequency in $\mathcal{H}$, rather than learning a mapping. The space of labs is much smaller than the space of symptoms or conditions, so we find that a frequency-based ranking is nearly optimal in practice.
4. Medications: As with labs, we rank by frequency for the same reasons.

Autocompleting Conditions

Documenting relevant patient history is often an arduous task for physicians in the ED. Doctors typically read a patient's triage assessment and then search through a patient's EHR on an ad-hoc basis to try and contextualize the current visit with the patient's background. In our dataset, there is a median of 65 distinct conditions mentioned in a patient's EHR, but on average, only 5 of these concepts are then documented in the ED clinical note. In addition, around a quarter of the patients in our dataset do not have any prior records on file; in these cases, doctors can guess relevant conditions to inquire about based on the triage text and chief complaint alone.

This leads to key model desiderata: first, we must be able to recover an intelligent ranking over concepts even in the absence of prior medical notes using triage information alone. Second, we seek to learn a single multilabel ranking over all possible model relevancy buckets in order to produce a globally calibrated model. Our model first learns a ranking over the coarse model relevancy buckets, and then recovers a ranking over individual condition concepts to mention in the note. We use a shallow, dual-branch neural network architecture to combine a context c_i consisting of a TF-IDF representation of the triage text T and a feature vector indicating the binary presence $\mathbb{1}[b \in \mathcal{H}]$ of each model relevancy bucket b in prior EHR notes $\mathcal{H}$. Each arm of the network is passed through a single dense layer with rectified linear unit activation, the two outputs of the dense layers concatenated, and then the combined embedding is passed into a final dense layer with sigmoid activation to provide a vector of estimates of relevancy for each bucket. We recover a term-wise ranking by sorting each term first by whether it appears in the EHR, then by the rank of its relevance bucket, and finally by its empirical frequency of occurring in the data to resolve ties. In this way, we create a single architecture that predicts $P(b|T, \mathcal{H})$, or the probability of b being relevant given the triage information and prior history, for all b simultaneously and thereby suggest conditions to document for patients both with and without a prior medical history. Training details for this architecture can be found in Section E.2.2. We also compare against three baselines:

1. *One vs. Rest Logistic Regression on Triage Text*: We build a model based solely on T. For each model relevancy bucket b, we estimate the $P(b|T)$ via a logistic regression model trained on a TF-IDF representation of T to predict if any term in b was mentioned in the corresponding clinical note. We randomly select notes without any mention of b to generate negative samples. To make a prediction for a given patient, we then rank relevance buckets b by $P(b|T)$. To recover a term-wise ranking, we sort each term first by the rank of its corresponding relevance bucket and by its empirical frequency in clinical notes.

2. *One vs. Rest Logistic Regression on Triage Text, EHR*: As above, we train a logistic regression model on T for each model relevancy bucket b. However, when predicting $P(b|T)$, we restrict ourselves to train on samples where b is mentioned in $\mathcal{H}$. That is, our model predicts the probability $P(b|T, \mathbb{1}[b \in \mathcal{H}])$. We assume

$P(b|T, 0) = \epsilon_b$ for a small but nonzero ϵ_b, or that if if a term does not appear in a patient's EHR, it is unlikely that it will be documented in the present note. To recover $P(b|T)$, we multiply by an empirically computed prior probability $P(\mathbb{1}[b \in \mathcal{H}])$ of each bucket being mentioned in the EHR. We recover a term-wise ranking using the same key as the previous method. The leak probability ϵ_b allows us to rank buckets that are not present in the EHR by their empirical probabilities alone, giving us predictive power for patients without any prior history.

3. *Augmented One vs. Rest Logistic Regression on Triage Text, EHR*: We experiment with feature-engineering approaches to include signals from the EHR in our model covariates. In particular, we augment the feature space with a representation D of how many days it has been since b was mentioned in the EHR, and compute $P(b|T, D, \mathbb{1}[b \in \mathcal{H}])$ via logistic regression. In order to force this input variable to conform to a normal distribution, we transform the delay times by assuming mentions follow a Poisson process and concluding that delay times should be exponentially distributed. We follow the same empirical reweighting and term-wise ranking procedure as in the previous model.

Autocompleting Present Symptoms

Based on discussions with clinicians as well as qualitative analyses within our slice of ED data, we find that the symptoms that a doctor asks a patient about and subsequently records are primarily rule-based. A chief complaint of dyspnea at triage-time, for example, might prompt the doctor to inquire about dyspnea (reaffirming that it is still a concern), chest pain, coughing, etc. Consequently, the models we develop for symptom autocompletion use only the chief complaint and triage vitals as covariates. We perform ablation tests with all of our models to confirm that adding in a bag-of-words representation of the triage text did not increase performance, and develop four schemes to map chief complaints and vitals to a ranking over symptoms:

1. *Empirical Conditioning on Chief Complaint:* For a given chief complaint c, we

empirically calculate $P(s|c)$ for each s in the set of symptoms S, and rank each symptom by this probability.

2. *Empirical Conditioning on Chief Complaint, Vital:* For a given chief complaint c and a list of vitals V, we calculate the single vital $v \in V$ that is most abnormal. Abnormality is defined as the percentile deviation from the population median of the vital value. We then encode v as a categorical variable $b(v)$ based on medical guidelines about the given vital (for example, heart rate vitals are placed into one of three buckets: `LOW HR`, `NORMAL HR`, and `HIGH HR`). Full details about the bucketization procedure can be found in the Section E.1.4. Finally, we empirically calculate $P(s|c, b(v))$ for each $s \in S$, and rank each symptom by this probability.

3. *One vs. Rest Logistic Regression:* For each symptom $s \in S$, we train a logistic regression model mapping the chief complaint and vital values to whether s appears in the ED note corresponding to that visit. Then, we rank the output probabilities for each symptom.

4. *One vs. Rest Naive Bayes:* For each symptom $s \in S$, we train a Naive Bayes classifier mapping the chief complaint and vital values to whether s appears in the ED note corresponding to that visit. Then, we rank the output probabilities for each symptom.

In practice, we find that the second scheme performs best and we use this for deployment. Comparative performance for these models is detailed in Section 8.4.

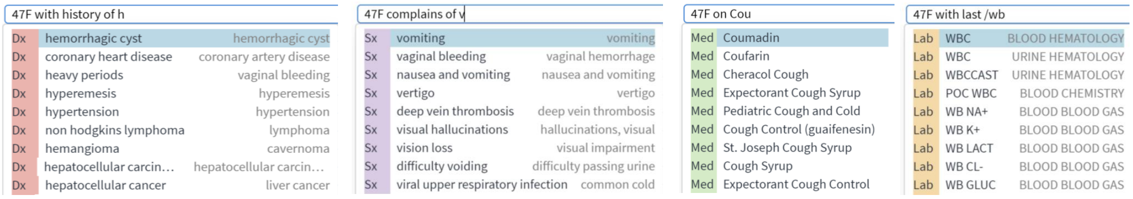
Autocompleting Labs and Medications

Autocompleting labs and medications is different from symptoms and conditions in a few marked ways. A patient's medical record contains structured information about prior lab tests and values, as well as medications and their dosages prescribed in the past. This is in contrast to symptoms and conditions which are almost always referenced in unstructured notes or free text. Concept disambiguation is less pertinent because there are structured representations of labs and medications, and there are already semi-structured lists of labs and medications that exist in clinical records. The primary value-add for physicians to tag a mention of a lab/medication in a note is instead to *enable immediate information retrieval.* Tagging `HCT`, for example, can prompt the visualization or insertion of a patient's hematocrit trend. We thus add lab and medication autocompletion to be thorough in our data collection, and use a frequency-based autocompletion for both data types.

Determining Autocompletion Scope and Type

There are two components to displaying autocompletion suggestions: (1) the *scope* of autocompletion, which determines when a clinician wants to document a concept; and (2) the *type* of autocompletion, which determines a ranking over whether the clinician wants to document a condition, symptom, lab, or medication. A potential approach to this problem is to build a sequential model predicting whether the next word typed will be a clinical concept and its corresponding type, but this requires significant client-side infrastructure to curb model latency– Gmail's Smart Compose system, for example, which surfaces dynamic suggestions of words to type from a neural language model, is only made possible via custom hardware and extensive system infrastructure [49]. We discuss this further in Section 8.5. To build a system which can run live, we instead adopt a rule-based approach.

We first define a default concept-type ranking per note section. For example, in HPI, the majority of documented content pertains to historical conditions and some current symptoms/medications, so the default ordering is `CONDITION, SYMPTOM, MEDICATION, LAB`. In contrast, in a Physical Exam section, clinicians document symptoms more than chronic conditions, so the default ordering is `SYMPTOM, CONDITION, MEDICATION, LAB`. We then establish certain key phrases to act as autocompletion triggers if they are likely followed by a clinical concept. We curate a list of common trigger phrases (e.g., `presents with`, `history of`) and map them to the concept type that follows them– `presents with` is mapped to `SYMPTOM`, and `history of` to `CONDITION`. Using these, we create a NegEx-inspired algorithm to predict both autocompletion type and



47F with history of h

Dx	hemorrhagic cyst	hemorrhagic cyst
Dx	coronary heart disease	coronary artery disease
Dx	heavy periods	vaginal bleeding
Dx	hyperemesis	hyperemesis
Dx	hypertension	hypertension
Dx	non hodgkins lymphoma	lymphoma
Dx	hemangioma	cavernoma
Dx	hepatocellular carcin…	hepatocellular carcin…
Dx	hepatocellular cancer	liver cancer

47F complains of v

Sx	vomiting	vomiting
Sx	vaginal bleeding	vaginal hemorrhage
Sx	nausea and vomiting	nausea and vomiting
Sx	vertigo	vertigo
Sx	deep vein thrombosis	deep vein thrombosis
Sx	visual hallucinations	hallucinations, visual
Sx	vision loss	visual impairment
Sx	difficulty voiding	difficulty passing urine
Sx	viral upper respiratory infection	common cold

47F on Cou

Med	Coumadin
Med	Coufarin
Med	Cheracol Cough
Med	Expectorant Cough Syrup
Med	Pediatric Cough and Cold
Med	Cough Control (guaifenesin)
Med	St. Joseph Cough Syrup
Med	Cough Syrup
Med	Expectorant Cough Control

47F with last /wb

Lab	WBC	BLOOD HEMATOLOGY
Lab	WBC	URINE HEMATOLOGY
Lab	WBCCAST	URINE HEMATOLOGY
Lab	POC WBC	BLOOD CHEMISTRY
Lab	WB NA+	BLOOD BLOOD GAS
Lab	WB K+	BLOOD BLOOD GAS
Lab	WB LACT	BLOOD BLOOD GAS
Lab	WB CL-	BLOOD BLOOD GAS
Lab	WB GLUC	BLOOD BLOOD GAS

Figure 8-3: Screenshots of contextual autocompletion tool for each autocompletion type. From left to right: (a) Conditions (b) Symptoms (c) Medications and (d) Labs. Trigger words before the tagged term affect the scope and type of the autocompletion. Clinical concepts with synonyms that match the typed text are listed with the synonym in black text and the more general concept name in gray.

scope by greedily matching triggers in the text [45]. A full algorithm sketch of this is included in the Section E.1.2, and screenshots of the scope and type prediction algorithms at work are shown in Figure 8-3.

While we rely on autocompletion scope and type prediction algorithms to guess where the user will insert a tagged term and the types of these terms, we support fallback data capture methods for when our algorithms fail. We do so in two ways. First, a user can start an autocomplete scope with a manual trigger. In addition, if the user does not type the manual trigger, we use an Aho-Corasick keyword detection algorithm to efficiently map exact string matches in the text with clinical concepts to our ontology [8]. Any matches are displayed as potential tags which doctors can manually confirm if desired. A screenshot depicting these backup data capture strategies can be seen in Figure E-2 of the Appendix. We analyze how often these mechanisms are exercised in practice below. Our rule-based algorithms have an average end-to-end latency of ≈ 0.2 milliseconds to make a prediction, which is well below the 100ms threshold for a response to feel instantaneous [193]. In contrast, making an API call to a shallow convolutional neural network for scope and type prediction takes upwards of 250ms in the absence CPU throttling, network overload, etc.

8.4 Results

A physician uses contextual autocompletion by naturally typing a note and either automatically or retroactively completing clinical phrases that are then rendered as tagged concepts. As in standard autocomplete, autocompleted concepts are filtered to those that match the user's typed prefix. We briefly describe the user experience of the tool with a screenshot in Figure 8-3, and examine how it reduces clinical documentation burden in practice.

8.4.1 Performance and Usability Metrics

Retrospective Evaluation on Clinical Notes

Before deploying our autocompletion models in a live setting, we evaluated the quality of our suggested rankings via retrospective annotation of the clinical notes we had on file. In particular, we measured performance broken down by concept type, as well as the efficacy of our autocompletion scope and type detection algorithms. We use two standard information retrieval metrics, the *mean reciprocal rank* (MRR) and *mean average precision* (MAP) to gauge the quality of our rankings (definitions in Section E.2.3). We compare the MRR of our contextual autocompletion tool rankings against two naive baselines: spell-based autocompletion (ranking terms alphabetically) and frequency-based autocompletion (ranking terms by frequency). Performance by MAP is detailed in the Section E.2.3.

To generate our evaluation set, we extract medical concepts from 25,000 clinical notes with the technique outlined in Section 8.3.3. Using the order in which concepts were suggested, we first measure MRR assuming our scope and type detection was perfect, broken down by the four concept types. Results are shown in Figure 8.1. We see the largest gain in using a contextual model for conditions, because the space of terms is large and the richness of the EHR greatly influences documentation. Within the contextual models for predicting prior conditions, the dual-branched neural network outperforms others primarily because it is predictive even for patients who did not have any history on file at the hospital. On the other hand, when documenting symptoms, a model that ranks symptoms by their empirical frequency (conditioning on the chief complaint and the most abnormal vital) performs best.

To quantify the ease of documentation using our autocompletion scope and type detection algorithm, we also measure MRR when typing HPI sections of notes. We focus on HPI notes as they contain a range of concept types (conditions, symptoms, medications, etc.) and also were the only note section we could reliably segment due to dataset limitations. On average, there are 6.8 documented clinical concepts per HPI section. Of the extracted clinical concepts in HPI sections, 46% of terms were autocompleted automatically without a manual trigger, and in 77% of those cases, we guessed autocompletion type correctly as well. As a result, the MRR of automatically-detected autocompleted terms is 0.35. Even in cases where the doctor is forced to insert a manual trigger to autocomplete a term, we still greatly decrease the documentation burden on doctors as shown in Figure 8.1. These manually-prompted scenarios can be mitigated as a doctor learns and adapts to the autocompletion triggers of the system, which we elaborate on in Section 8.5.

Documentation in the Wild: Live Evaluation

Because the primary goal of this tool is to improve documentation efficiency, we also define the *keystroke burden* as the number of keystrokes the clinician needs to type until he/she autocompletes and inserts a desired term. This usability metric inherently encompasses the quality of our information retrieval in its calculation while also incorporating real-world behavior—there may be a delay between a term being suggested first and when a clinician actually autocompletes the term. We compare keystroke burden between a contextual model and no autocompletion in Figure 8.2. In our live evaluation, a single physician wrote 40 notes using our system over two shifts. In practice, an average of 8.38 terms are tagged per note, and we reduce overall keystroke burden for these clinical concepts by approximately 67%, with clear gains in using our model irrespective of note section or concept type. 53% of the tagged clinical concepts were autocompleted without a retroactive label. Moreover, 96% of these terms were automatically prompted (as opposed to the user manunally prompting the

(a) Comparison of MRR between contextual autocompletion models

Model Type	MRR ↑
Conditions	
One vs. Rest Logistic Regression on T	0.09 ±0.02
OvR LR on T, EHR	0.15 ±0.02
Augmented OvR LR on T, EHR	0.17 ±0.01
Dual-branched neural network	**0.28** ±0.01
Symptoms	
Empirical Conditioning on Chief Complaint	0.39 ±0.01
Empirical Conditioning on Chief Complaint, Vital	**0.42** ±0.01
One vs. Rest Logistic Regression	0.16 ±0.01
One vs. Rest Naive Bayes	0.27 ±0.02

(b) Comparison of MRR across autocomplete types

Model Type	Autocomplete Type		
	Spell	Frequency	Contextual
Conditions	0.01 ±0.001	0.08 ±0.01	**0.28** ±0.01
Symptoms	0.05 ±0.001	0.27 ±0.01	**0.42** ±0.01
Labs	0.01 ±0.001	0.40 ±0.01	N/A
Medications	0.02 ±0.001	0.02 ±0.001	N/A
Overall	0.01 ±0.001	0.19 ±0.03	**0.29** ±0.05

Table 8.1: Retrospective Evaluation of MRR using Contextual Autocompletion. We report average MRR (±95% confidence interval of the mean) for each of our learned contextual autocomplete models, and compare our best models (dual-branched neural network for conditions, empirical conditioning on the chief complaint and most abnormal vital for symptoms) to spell-based and frequency-based baselines, both for specific concept types as well as overall using our scope and type prediction algorithms. Calculated across 25,000 visits.

autocomplete), indicating our scope and type detection had high recall even when doctors had not yet adapted to the system. For 77% of the terms tagged without a retroactive label, we also predicted the clinical concept type correctly.

Subset	Autocompletion Type	
	None	Contextual
Overall	11.85 ±1.94	4.32 ±0.43
By Note Section		
History of Present Illness	12.36 ±2.16	4.57 ±0.87
Past Medical History	11.41 ±2.09	2.94 ±0.68
Medical Decision Making	10.27 ±3.18	4.08 ±0.49
By Concept Type		
Conditions	13.08 ±1.72	4.34 ±1.49
Symptoms	8.5 ±2.18	4.53 ±1.00
Labs	10.33 ±5.76	2.06 ±0.88
Medications	9.27 ±1.97	4.27 ±1.34

Table 8.2: Live Evaluation of Contextual Autocompletion Models. Mean keystroke burden for autocompleted concepts (± 95% CI from mean), measured across 40 notes written live by a single physician over two shifts. Performance is also broken down by note section, as well as concept type.

8.4.2 Autocompletion Sensitivity Analysis

Concept frequency influences the efficacy of our contextual autocompletion model of conditions. The biggest wins in the model occur with the group of conditions in the middle of the frequency distribution. `Renal insufficiency`, for example, is an infrequent but not rare term that will almost certainly be documented in a note if it appears in the patient's history. The symptom contextual autocompletion model, on the other hand, is generally agnostic to concept frequency because the space of symptoms is much smaller and the distribution of symptoms is less skewed than that of conditions.

In addition, the presence of prior medical history has significant impact on contextual autocompletion performance for conditions. As shown in Table 8.3, we see greater reduction in documentation burden if the patient has prior EHR. However, our contextual model and a frequency-based autocompletion model perform similarly for concepts that are not mentioned in the EHR despite the person having some prior medical history. This can largely be attributed to the inherent bias of our ranking scheme, which preferentially orders terms mentioned in the EHR above those that are not.

Mean Keystrokes Saved per Condition Concept			
	Uncommon Concepts	Median Concepts	Common Concepts
With no past EHR at hospital	0.63 ±0.42	0.81 ±0.50	0.47 ±0.20
With prior mention of concept in EHR	2.64 ±0.65	2.02 ±0.38	1.40 ±0.16

Table 8.3: Number of keystrokes saved by our contextual model compared to a frequency-based baseline (±95% CI of the mean) for conditions. Performance was stratified by concept frequency (by terciles) and by available medical history.

Concept	**Most Predictive Triage Tokens**	**Most Predictive Model Relevancy Buckets**
Dementia	dementia, abrasions, fell, home, fall, neuro, son, ...	dementia, neurodegenerative diseases
Bronchitis	pna, pneumonia, cough, sob, hemoptysis, sputum, ...	pneumonia, chronic lung disease
Prostate cancer	ca, mass, chemo, lymphoma, melanoma, cll, tumor, ..	cancers, prostatectomy
CHF	chf, chest, sob, cp, cough, syncope, fall, ...	heart failure, heart attacks, hypertension, afib
Diabetes	bs, fsbs, glucose, iddm, sugars, toe, finger, ...	diabetes, hyperlipidemia, diabetic neuropathies, gastroparesis

Table 8.4: Predictive features for selected condition concepts, using a linear approximation to our contextual model for conditions. Inputs to the model are a TF-IDF representation of the triage text as well as the presence of coarse-grained model relevancy buckets in a patient's prior medical record, as defined in Section 8.3.2.

8.4.3 Interpreting Contextual Autocompletion of Prior Conditions

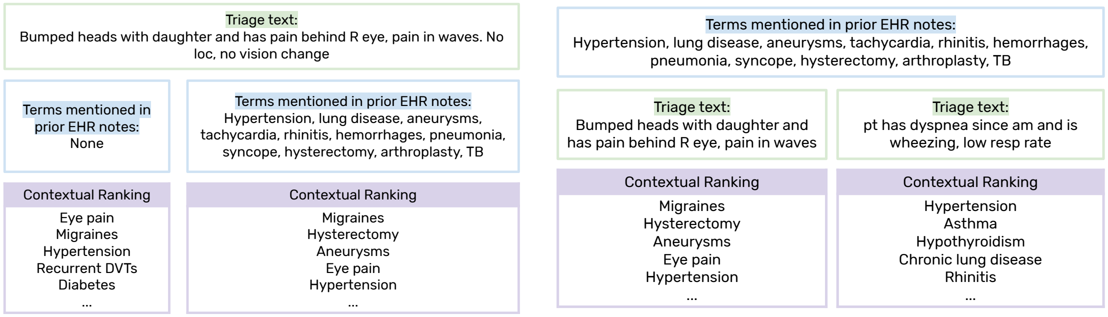
Because our contextual model for conditions learns a ranking from a representation of the triage text and medical history, it is naturally more sensitive to changes in input than our contextual model for symptoms. Here, we dig further into what drives model predictions.

Performance By Concept

Our multi-label model predicts the binary relevance of each model relevancy bucket. To better interpret relevancy predictions on a per-bucket level, we approximate our model for a specific relevancy bucket b with a linear function of the inputs. This is done by fitting a L_1-regularized linear approximation between the features and the logits generated by the model for bucket b to surface highly-weighted features [162]. In Table 8.4, we provide examples of the top-weighted positive features in the linear approximations to models for five selected concepts. Overall ranking performance by MRR for these concepts is in Figure E-3 of the Appendix. Interestingly, while all of the chosen concepts relied on medically meaningful tokens present in the triage text, the linear models for diabetes and congestive heart failure both used the presence of many model relevancy buckets, whereas the other three concepts only relied on a few. This is likely because the model always relies on triage text but can give predictions even in the absence of prior medical history, and as the linear approximation to our model encourages sparsity, only highly predictive model relevancy buckets will be chosen as features. A frequency-based baseline outperforms our learned model only for extremely common conditions like hypertension and diabetes.

Qualitative Evaluation & Readability

We qualitatively evaluate rankings over conditions to better understand model decisions. As can be seen in the selected examples in Figure 8-4, both the presence of EHR notes as well as specific types of words mentioned in the triage note can have great impact on the rankings, which are much more context-specific than frequency-based rankings. Chronic conditions mentioned in a patient's medical history are highly ranked even if they are not directly related to the present medical context, because they are likely to be documented regardless. For example, in Figure 8-4a, two patients have identical triage text but different medical histories—consequently, `hysterectomy` is highly ranked for one. Of course, the triage note still governs the overall theme of the most highly ranked terms; in Figure 8-4b, two patients with identical medical


Triage text:
Bumped heads with daughter and has pain behind R eye, pain in waves. No loc, no vision change
Terms mentioned in prior EHR notes:
None
Terms mentioned in prior EHR notes:
Hypertension, lung disease, aneurysms, tachycardia, rhinitis, hemorrhages, pneumonia, syncope, hysterectomy, arthroplasty, TB
Contextual Ranking
Eye pain
Migraines
Hypertension
Recurrent DVTs
Diabetes
...
Contextual Ranking
Migraines
Hysterectomy
Aneurysms
Eye pain
Hypertension
...
Terms mentioned in prior EHR notes:
Hypertension, lung disease, aneurysms, tachycardia, rhinitis, hemorrhages, pneumonia, syncope, hysterectomy, arthroplasty, TB
Triage text:
Bumped heads with daughter and has pain behind R eye, pain in waves
Triage text:
pt has dyspnea since am and is wheezing, low resp rate
Contextual Ranking
Migraines
Hysterectomy
Aneursyms
Eye pain
Hypertension
...
Contextual Ranking
Hypertension
Asthma
Hypothyroidism
Chronic lung disease
Rhinitis
...

(a) Effect of patient history on contextual rankings.

(b) Effect of triage note on contextual rankings.

Figure 8-4: Case Studies of Rankings over Conditions

histories but differing chief complaints have vastly different context-specific rankings.

8.5 Discussion

The contextual autocompletion tool we have outlined harnesses the power of machine learning to encode information about medical contexts, and then uses this to suggest terms to document to clinicians. Medical professionals who utilize this tool can not only document terms more easily and save valuable time to interact directly with patients, but also can create clean annotations of clinical text in a novel manner. These annotations can be used to provide disambiguation between overloaded terms, clarify associations between medical concepts, and generate large-scale EHR datasets for future innovation. All of the medical ontologies built for this chapter map to UMLS, making our contextual autocompletion tool translatable to other clinical centers with minimal modification. The ablation tests we carried out show that using a few features (primarily representations of medical histories) can result in performant predictive models for documentation. This is a critical advantage of our system because EHR data is often very sparse—patients can enter the ED with no prior medical history, yet we can still glean information from the triage assessment to represent a patient state. Our strategies also obviate the need for complex data imputation schemes.

Our system provides automatic and natural autocompletion of a clinical concept when our scope detection algorithm is accurate, and users may need to resort to a manual trigger in other cases. This is not a major hindrance based on our evaluation criteria, and we significantly reduce documentation burden even if the manual trigger is used. However, we note that more complex model classes (e.g., recurrent neural networks/sequential learning models) may be better at scope prediction, at the expense of client-side latency. That being said, using manual triggers for autocompletion, however, can also establish consistent system behavior for physicians, and create notes that are concise. As an example, one clinical note in our dataset began with the phrase `patient has a history of abdominal pain which seems recurrent`, whereas our system would autocomplete to `patient has a history of chronic abdominal pain`. While we we do not explore these schemes in this chapter, further study of this trade-off can be found in section 5.2 of Gopinath [89].

We propose future directions to build on and continue this work. The first is to better integrate key semantic modifiers into our tool. As an example, doctors often document the absence of symptoms (e.g., `no fever`) to aid in a differential diagnosis. While we can use rule-based approaches to retrospectively attach negation modifiers to tagged medical concepts, future work should seek to fuse modifier capture with the UI, which is explored in the following chapter. Clinicians might also type a term that refers to someone other than the patient (e.g., `family history of diabetes in mother`), and we should automatically learn that the concept `diabetes` refers to a third-party rather than the patient.

In addition to facilitating semantic modifier capture, a next iteration should dynamically update suggested terms to document using already-tagged terms in the note. Tagging `atrial fibrillation`, for example, might indicate that there is a high likelihood of the doctor typing an anticoagulant next. Using live data collected from the deployed tool, we can use early drafts of a clinical note to influence the medical context for later autocompletion suggestions. We can also clarify patterns of redundant data entry by examining where the same underlying medical concept is repeated in the note, with the eventual goal of learning and auto-inserting necessary repetitious documentation. These dynamic updates introduce a significant latency on the client-side UI to perform online inference as words are typed, so this may not be feasible for all systems and thus we did not consider it in this first iteration.

Finally, further work should be conducted to understand the possible bias induced by terms that appear higher in the autocomplete ranking, if any. As autocompletion of longer phrases becomes increasingly feasible, studying this in greater detail will be paramount. New paradigms will be required to do wider AI-assisted documentation safely, as we must consider veracity of suggested text, grounded support for that text, and a potential loss of agency among clinicians, as observed previously in Chapter 4.

8.6 Conclusion

EHRs have introduced significant burden on physicians, and to adapt, doctors have resorted to using overloaded jargon that then renders clinical notes unusable for downstream clinical care. The lack of clean labels for unstructured text also inhibits how we can utilize machine learning techniques to transform healthcare. There is a real need to modernize and exploit the information hidden within notes without interrupting the clinical workflow. While our contextual autocompletion tool can reduce documentation burden and curate clean data for machine learning purposes, it also opens the possibility of reforming clinical documentation practices to make notes more understandable to humans and algorithms alike. Fundamentally, live-tagging of medical concepts enables unprecedented changes to EHR design. By integrating machine learning methodologies into documentation practices, we can usher in a new era of EHRs that assist rather than impede physicians.

Chapter 9

Integrating Automated Information Retrieval into EHRs

Acknowledgement of Co-authors I am the third author on this paper, following first author Luke Murray and my Master's supervisee Divya Gopinath. Luke headed development of the front-end and user interface components described in this work, and Divya headed development of the information retrieval components described here. As with Chapter 8, Divya and I spoke nearly daily about this work, and Luke and I collaborated at a less frequent cadence. I was highly involved in the writing, analysis, and evaluation presented in this chapter.

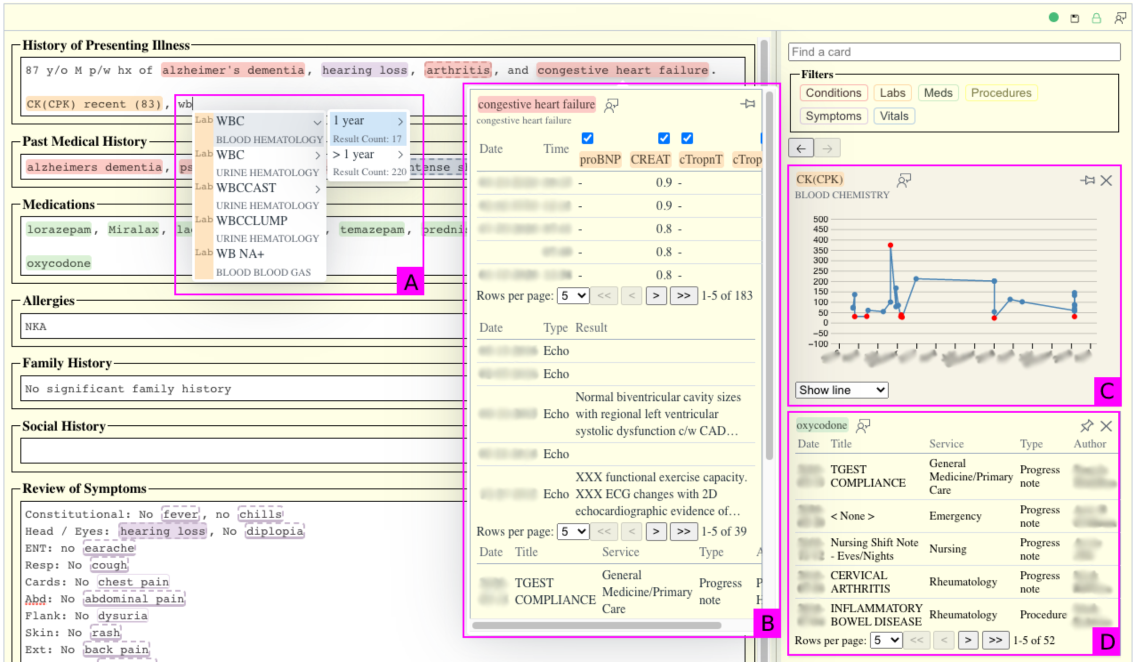
9.1 Introduction

In this chapter, we describe how the paradigm of prospective data capture can facilitate automated information retrieval in EHRS. EHRS were adopted in the hope that they would improve quality of care, save time, support collaboration and data sharing, and prevent clinical errors [59, 24, 77, 241]. However current EHR platforms have largely failed to achieve these goals. Studies of EHR adoption have shown both positive and negative effects [167, 140], but clinicians now spend more time navigating EHRs than physically communicating with patients and EHR usage is a leading cause of physician burnout and stress [170, 251, 180].

Despite being laborious to create, well-written clinical documentation is invaluable. At their best, cogent clinical narratives can help clinicians understand a patient's case [204, 167], function as a powerful communication method between clinicians [56], and serve as learning tools to improve future care practice [24]. But EHRs rarely achieve this and arguably interfere with it. The issue lies in the fragmentation among views in the EHR for the two processes underlying the clinical workflow: (i) information retrieval and data exploration over a patient's history and (ii) information entry. Because structured and unstructured data can be hard to reconcile, EHRs often store and display information in separate pages or windows, and physicians have to synthesize the patient narrative by navigating across a variety of sources [229, 7]. This creates increased cognitive burden to discover unstructured information, and studies have shown that clinicians spend more time reading past notes than doing any other activity in the EHR [39]. Further, the fragmented interfaces hinder comprehensibility and necessitate frequent task-switching [326, 167, 56]. To avoid this context switching, clinicians have developed coping mechanisms such as copying from previous notes or using autofill techniques for naive pre-population of text [100, 166, 230]. Unfortunately, indiscriminate use of these auxiliary functions causes documentation to become bloated, making it difficult for clinicians to parse important clinical information, and potentially even propagating errors [275, 230, 105, 295].

9.1.1 Contribution

In this chapter, we propose a novel documentation system for EHRs, MedKnowts, which passively assists clinicians by seamlessly integrating an editor for clinical documentation with a *concept-oriented view*[319] of the patient's medical history. MedKnowts provides contextual autocomplete (Fig. 9-1A) for clinical terms (e.g.,conditions, symptoms), saving precious documentation time. The autocomplete works without a trigger character—so it does not disrupt the prior documentation workflow—and displays options for structured data entry (e.g., lab values) as the user types, removing the need to memorize content importing phrases. When autocomplete is not used, we employ keyword matching, which we call *post recognitions*, to automatically identify



History of Presenting Illness: 87 y/o M p/w hx of alzheimer's dementia, hearing loss, arthritis, and congestive heart failure.
CK(CPK) recent (83), wb

Autocomplete:
- WBC — BLOOD HEMATOLOGY
- WBC — URINE HEMATOLOGY
- WBCCAST — URINE HEMATOLOGY
- WBCCLUMP — URINE HEMATOLOGY
- WB NA+ — BLOOD BLOOD GAS

- 1 year — Result Count: 17
- > 1 year — Result Count: 220

Past Medical History: alzheimers dementia, ps… …ntense s…

Medications: lorazepam, Miralax, la… …, temazepam, predni…
oxycodone

Allergies: NKA

Family History: No significant family history

Review of Symptoms:
Constitutional: No fever, no chills
Head / Eyes: hearing loss, No diplopia
ENT: no earache
Resp: No cough
Cards: No chest pain
Abd: No abdominal pain
Flank: No dysuria
Skin: No rash
Ext: No back pain

congestive heart failure
congestive heart failure

Date	Time	proBNP ☑	CREAT ☑	cTropnT ☑	cTrop
[redacted]	[redacted]	-	0.9	-	
[redacted]	[redacted]	-	0.9	-	
[redacted]	[redacted]	-	0.8	-	
	[redacted]	-	0.8	-	
[redacted]	[redacted]	-	0.8	-	

Rows per page: 5 << < > >> 1-5 of 183

Date	Type	Result
[redacted]	Echo	
[redacted]	Echo	
[redacted]	Echo	Normal biventricular cavity sizes with regional left ventricular systolic dysfunction c/w CAD…
[redacted]	Echo	
[redacted]	Echo	XXX functional exercise capacity. XXX ECG changes with 2D echocardiographic evidence of…

Rows per page: 5 << < > >> 1-5 of 39

Date	Title	Service	Type	A
[redacted]	TGEST COMPLIANCE	General Medicine/Primary Care	Progress note	P H

CK(CPK)
BLOOD CHEMISTRY

oxycodone

Date	Title	Service	Type	Author
[redacted]	TGEST COMPLIANCE	General Medicine/Primary Care	Progress note	[redacted]
[redacted]	< None >	Emergency	Progress note	[redacted]
[redacted]	Nursing Shift Note - Eves/Nights	Nursing	Progress note	[redacted]
[redacted]	CERVICAL ARTHRITIS	Rheumatology	Progress note	[redacted]
[redacted]	INFLAMMATORY BOWEL DISEASE	Rheumatology	Procedure	[redacted]

Rows per page: 5 << < > >> 1-5 of 52

Figure 9-1: The MedKnowts interface containing sections of the clinical note on the left, and an integrated sidebar on the right. The user is typing `WBC` and triggering autocomplete (A). Detail text in the autocomplete is used to differentiate clinical terms and provide additional context such as result counts. The card for the most recent identified term, `CK`, is displayed in the preview pane (C) with values displayed as a line chart, and abnormal values highlighted in red. The preview pane history can be navigated using the backwards and forwards buttons at the top of (C). Below the preview pane the doctor has pinned a card for `oxycodone` (D), which displays note snippets relevant to `oxycodone`. A search bar at the top of the sidebar can be used as an alternative method to add cards to the preview pane. A transcluded card for cardiac conditions (B) shows labs, cardiology reports, and note snippets relevant to congestive heart failure in a single interface.

clinical terms as the clinician types. Both auto-completed and post-recognized terms are transformed into structured interactive elements which we call *chips*. We leverage this structure for live semantic highlighting that enables easier parsing of long notes and for automatic population of repetitive text fields, easing documentation burden. Therefore, MedKnowts retains some of the benefits of structured data entry, while still allowing users the flexibility of natural language to describe the subtleties of complex patient narratives.

Further, we use the structured data to automatically surface *information cards* in an attached preview pane (Fig. 9-1C) as the doctor types. Proactively displayed cards provide concise summaries of relevant medical history, reducing the context-switching required to synthesize a note. Each card is a concept-oriented view [319] such that information is grouped by underlying concept (e.g., the labs, medications, and notes related to a condition) rather than by data modality (all medications at once). Concept-oriented views have been shown to help physicians work faster and make fewer errors [244]. In addition to the automatically surfaced cards, chips embedded in the note and in cards serve as links to related cards, providing direct access to the relevant medical history from the note context and other cards. Cards can be surfaced in-line by hovering on a chip (Fig. 9-1B) or in the preview pane by clicking on a chip. This provides an additional avenue for contextual information retrieval without dividing attention between views. Finally, cards can be pinned to an attached sidebar (Fig. 9-1D), which persists the card to a view shared by the clinical care team, allowing for easier bookmarking, collaboration, and data sharing without directly copying to contribute to note bloat.

We present the following contributions to enhance the EHR note taking experience:

- We provide passive and automatic methods to insert and disambiguate clinical terms as the note is written and transform them into chips—interactive, structured elements which provide information scent about recognized vocabulary, semantic highlighting, access to inline documentation, and contextual information retrieval. We therefore retain benefits of structured data entry without sacrificing the flexibility or ease of natural language.

- We augment the EHR note-taking interface with a shared sidebar to which clinicians can pin cards. Each card presents a concept-oriented view for a particular clinical term. The sidebar provides clinicians with a shared and persistent space, integrated with the documentation interface, where they can add and remove cards. It thereby situates, beside the semi-structured note, a collaborative, customizable, and context-specific view of structured data in a patient's medical record.

- We proactively display a preview card of the most recently identified concept which updates as the user types. The preview card provides a consistent passive display of detailed information immediately relevant to the clinician's current decision making context, reducing the need for the physician to manually forage for information.

- We present findings from a year long iterative prototyping and design process and a one month evaluation with four medical notetakers.

We implemented these designs in a prototype system which we deployed live among scribes in an Emergency Department (ED) at a Level I trauma center and tertiary, academic, adults-only, teaching hospital. Our system was designed over the course of a year, in collaboration with an emergency physician with over a decade of experience designing and deploying EHRs, and with ongoing feedback from stakeholders including scribes, medical students, and physicians. In practice, scribes found MedKnowts easy to use with a quick learning curve and and indicated that they would use it frequently. Further, they found the features of MedKnowts well-integrated, saving them time over their previous workflows both for documentation and information retrieval.

9.2 Related Work

9.2.1 Information Capture

Early EHRs were expected to transform clinical care by transitioning medical records from manually-organized and paper-based to automatic and digitized [16]. Many early EHRs were built around forms and structured data entry in order to capture structured records, but few modern EHR systems retain these designs [226]. Structured data entry is far more cumbersome and time consuming to input than unstructured text [267, 226, 169]. Clinicians prefer recording information with unstructured narrative [279, 220] because of the increased expressivity of free-text [297, 122]. However even clinicians who want the flexibility and efficiency of free-text when documenting information prefer structure and standards when revisiting old notes to parse the patient's medical history [226, 122]. MedKnowts lets clinicians seamlessly access and capture structured patient information and clinical terms while writing free text narrative. MedKnowts additionally synthesizes the existing patient medical record into concept-oriented cards which provide the clinician with a standardized and structured view of data extracted from a pre-existing EHR system.

Automatic Term Recognition

Most clinical recognition systems are designed for post-processing rather than real-time analysis [237, 54]. They extract structured information from unstructured narrative and free text after it has been authored [122, 226]. Systems such as *Doccurate* [262] have been designed to validate, augment, and visualize post-processed labels but few systems close the loop and enable clinicians to take advantage of identified structure in the medical note during the process of documentation [226].

Of the few proposed EHR paradigms that do implement real-time entity recognition during notewriting, they either fail to map to standard clinical ontologies [27], neglect to use this structured data capture to support clinical decision-making [122], or do not provide concept disambiguation (Fig. 9-4) which is crucial given the overloading of medical terminology and limited accuracy of post-hoc clinical concept recognition [237, 27, 122, 54]. *Active Notes* [298] inspires the design of several features in MedKnowts such as tagging clinical concepts and displaying related information in an attached sidebar. However *Active Notes* requires users to manually initiate data queries and tag concepts with a hot key, and does not visually distinguish clinical vocabulary until it is tagged, making it hard for clinicians to learn the recognized vocabulary. In contrast MedKnowts is designed to passively and automatically assist users without active participation. MedKnowts provides live semantic syntax highlighting for clinical terms indicating concept type, negations, and potential ambiguities; and automatically transforms autocompleted and post-recognized clinical terms into interactive chips which can be used to resolve ambiguities, and view relevant patient information inline as a tooltip or persisted in an integrated sidebar.

Structured Data Capture

Many modern EHRs support multiple modalities for inserting structured data into the note [166]. Some tools support carry-forward techniques where data is copied or paraphrased from previous notes [101]; others let clinicians insert structured values into the note by clicking in the patient's history or typing special characters to trigger macros [299, 230]. Still others require the user to specify the template structure using a complex interface of forms [18].

MedKnowts differs from previous carry-forward techniques [230] by autofilling using information captured earlier in the note, rather than limiting autofill to information that appears in the patient's prior medical record. This is particularly pertinent to documentation in an ED environment, since clinicians often have to repeat information within the same note in order to meet regulatory and billing requirements, and previous notes may not be applicable to the current visit, let alone exist.

MedKnowts supports structured data capture for clinical terms (conditions, symptoms, medications), lab results, and vital signs with a machine learning-driven autocomplete interface based on Gopinath et al. [90]. The autocomplete interface displays completions of clinical terms as the user types, which provides information scent for the available clinical vocabulary. Structured data capture is a common feature in

EHRs often referred to as *dotphrases* because the data is conventionally inserted with a phrase that starts with a period (e.g., `.meds`) [230]. MedKnowts differs from previous systems because the structured data insertions do not require a trigger character or memory of content-importing phrases. Trigger characters were unpopular in our deployments, since they require foresight to enter and knowledge of valid phrases.

Additionally, structured data templates, documented in Rule et al. [230], work well in medical specialties such as ophthalmology, where many standard structured measurements are taken before the patient sees the doctor. However, in clinical settings such as the ED, the vast majority of structured data entry opportunities are contextually dependent on information needs arising after the clinician begins documentation. Thus, our more fluid workflow for structured data insertion within narrative text is an important extension to Rule et al.'s structured templates.

9.2.2 Information Fragmentation in EHRs

Studies of EHR usage have shown that separation of documentation interfaces from patient data cause clinicians to frequently task switch, creating cognitive overload and increasing the likelihood of clinical errors [167, 202, 7]. Some previous EHR systems attempt to resolve this by presenting the entire medical record next to the documentation interface in complex interface of tabs, lists, and tables [122, 68]. These interfaces are hard to parse, require manual navigation, and leave the complex work of synthesizing data from across the medical record to the clinician [167]. Other EHR systems, such as the one in use at the hospital in which we deployed, provide dashboards summarizing high value information next to documentation [7, 208]. In an ED these summary displays rarely include all the information clinicians need to access throughout the course of a visit. Still other research systems allow users to interactively filter a view of the patient's medical record to display data relevant to a particular concept [299, 241, 110]. These systems allow users to filter by one concept at a time and do not persist the data for later reference.

MedKnowts lets clinicians access a curated subset of the medical record, displayed as a collection of concept-oriented cards. Each card provides a succinct display of high value information curated for a single clinical concept. The card relevant to the most recently recognized term is automatically displayed next to the note in a preview pane, providing a passive stream of relevant information to the clinician. Previous work has shown that clinicians are much less likely to perform manual actions to see information [313]. Cards can also be manually pinned to the sidebar where they can be seen by all users working on the note. Pinned cards act as a persistent and shared collection of data which is particularly pertinent to a given patient's context.

9.2.3 Problem-oriented Medical Records

In the early 1970s Weed proposed the notion of problem-oriented medical records [290]. In the problem-oriented medical record, all information is organized around patient problems. Problem-oriented medical records were designed to reflect the way the physician thinks [268], but did not survive. A major reason for their failure is that they require physicians to enter and maintain data organized around problems—often requiring multiple steps to input a single piece of data, while competing chronologically-oriented medical records offered unstructured text entry which was lightweight and fast in comparison [267].

Problem-oriented medical records (POMR), problem-oriented views, and concept-oriented views are very similar but have slight distinctions. Problem-oriented medical records refers to original idea proposed by Weed [290] to organize medical records around a problem list. Problem-oriented views (POV), introduced by Buchanan [36], dynamically generate problem-oriented displays of information from a traditionally organized medical record. POVs do not require the user to input information organized around problems. POVs place the the burden of organizing information around problems on the computer not the user. Concept-oriented views (COV) introduced by Doré [64], are an extension of POVs to all concepts not just problems.

9.2.4 Information Foraging Theory

Information foraging theory draws parallels between how humans hunt for information and how animals hunt for food—in particular, it identifies that users rarely find information in a completely linear process. Instead, useful information often appears in patches for which the user must forage, using clues in the user interface referred to as information scent [210].

Previous research into information foraging theory in EHRs highlights that the value of clinical information is not intrinsic but rather dynamic and task-specific [86]. Information that is relevant and important for one patient during one visit may not be relevant or important for another patient or in another clinical context. MedKnowts presents a consistent stream of context-specific cards in a preview pane. Each card is analogous to an information patch, and the user can quickly determine if the card is worth foraging in and exploiting by reading the card title or scanning the card content which is consistent across cards. If a card is useful, the user can exploit the information patch by persistently pinning the card to their sidebar.

MedKnowts encodes information scent within the documentation interface by providing semantic syntax highlighting for clinical terms in the form of chips. Terms are colored based on their concept type, whether or not they are negated, and whether or not they collide with other terms—these clue the user into how we have inserted structure into the note and what downstream information benefits to expect. Small visual indicators next to clinical terms, and detail text in the UI provide additional information scent and inform clinicians about whether a card is likely to contain information from the patient's medical record. Users can easily navigate between information patches. Users can navigate to cards by clicking or hovering on clinical terms embedded within both cards and the note taking interface, or by searching for a clinical term in the sidebar.

9.3 Environment

9.3.1 Clinical Workflow

While MedKnowts was designed for use in a particular hospital's ED, here we describe a high-level clinical workflow that is generally common across EDs. During a typical day in a hospital ED, clinicians may evaluate, treat, and document up to 35 patients. The note is used for various purposes: as a tool for communication and collaboration between present and future clinicians; as a document of the evidence-based decision making process the clinician utilizes to construct a care plan; and as a record for legal and reimbursement purposes [59, 44]. Before the clinician evaluates a patient, a triage nurse first prioritizes a patient, taking vital signs, assigning a chief complaint, and writing a brief triage note. The clinician then evaluates the patient and reviews the patient's prior medical record. As in almost all healthcare settings, time is limited and must be balanced between bedside care and reviewing the patient medical record. The main sections expected in the final documentation then closely mirror the underlying clinical workflow after triage [44]:

History of Present Illness (HPI). The HPI serves as a chronological narrative of the patient's reason for the visit, including the presence, onset, severity, and duration of symptoms. Additionally, it involves surfacing medical history that may be relevant for contextualizing the patient's condition. Unlike in specialties that provide longitudinal care, emergency visits are episodic and unscheduled; emergency physicians are often meeting a patient for the first time, forcing them to quickly synthesize a patient's medical background from various sources, including past medical records.

Review of Symptoms (ROS). The ROS contains an inventory of symptoms, documented per body system (e.g., cardiovascular, gastrointestinal). Information from the HPI is often repeated here.

Medical Decision Making (MDM). MDM is the complex process by which the clinician reaches a diagnosis and treatment plan. Within the MDM section, physicians need to enumerate the differential diagnosis, consider risks associated with various diagnostic and treatment options, and settle on the labs, tests, medications, and scans that must be conducted as part of the workflow.

The sections above provide a comprehensive view of the patient's visit by corresponding to the systematic and thorough process behind patient evaluation and management. However, there is often overlapping information in the sections above due to billing requirements [55], e.g., the ROS may include symptoms that were already mentioned in the HPI, the MDM often contains elements of the past medical history, leading to complaints of excessive, often repetitive data entry [138].

9.3.2 A Variety of Documentation Processes

The documentation process described here is based on observations at the ED in which MedKnowts was deployed. Some aspects of this process, such as the use of scribes, may not generalize to other EDs. There is marked inter- and intra-provider variation in the processes to reach the final documentation based on individual clinician preferences, resources, and schedules. Some clinicians write the majority of notes after their shift, jotting details during to jog memory later. In addition to the final note, there exists an additional *Clinician Comment* box which can be used for such intermediate thoughts, and is often additionally used as scratch space between members of the care team (e.g., an attending physician, a resident, a medical student) that are not part of the medical record. Others choose to write the majority during the shift, only revisiting the notes to make small edits and submit their notes to the official record.

On another dimension, alternatives to keyboard text entry include (i) the use of voice dictation software and (ii) the employment of a scribe. Scribes shadow the clinician, recording what they observe during patient encounters as well as discussions with other clinicians, and drafts notes for each of the patients that the clinician is seeing. These notes are then handed over to the clinician, who will edit and augment to prepare the note for official recording in the patient's medical record. Since a lot of information communicated during the visit is irrelevant to the patient's care, the scribe acts as a filter that determines, documents, and relays clinically-relevant information. Experienced scribes may even search and synthesize the patient's past medical records themselves. Because scribes were already writing notes at the ED MedKnowts was deployed in, they were the target subjects for our study. However in other hospitals where clinicians act as their own scribes, the clinicians would be the target users. Voice dictation software can be used as an alternative text entry method when scribes are not available. But voice dictation does not fulfill other roles the scribe performs in the clinical workflow. In this study, due to incompatibilities in the deployed commercial dictation software, we specifically focused on scribe-physician workflow. However, we note that interaction with dictation software is an infrastructural challenge and not a fundamental obstacle to using our system.

9.3.3 Study Environment

The study described in this chapter was performed within a single Level I trauma center and tertiary, academic, adults-only, teaching hospital which provides care for 55,000 patients per year. The existing deployed web-based EHR was custom developed at the institution, but uses a commercially available documentation module. The study was approved by our institutional review board with a waiver of informed consent.

MedKnowts was developed through prototypal deployments over the course of a year, during which a clinician and the clinician's scribes used the tool as their predominant note system. We report on lessons learned from the iterative prototyping process, as well as usage data collected from a one month long deployment at the end of the year.

9.4 Design and Implementation

The overall goal of our system is to reduce the effort clinicians must invest in retrieving information from the EHR, synthesizing that information into knowledge, and recording it into patient notes. We do so via a combination of interacting features:

1. We use *autocomplete* as well as *post recognition* to recognize meaningful concepts from a large, standard medical ontology. Autocomplete can save users keystrokes. More importantly, these standard concepts provide an indication of the problem

the clinician is addressing for the current patient and are inserted as structured *chips*.

2. We use the recognized concepts to pre-populate other portions of the note that require duplication of that information, relieving clinicians and scribes of that burden.

3. We introduce a *preview pane* and persistent *sidebar* for delivery of standardized *cards* of contextual information relevant to recognized concepts. When a concept is recognized, the relevant card is automatically introduced in the preview pane, proactively providing clinicians with information they are likely to need to address the problem whose description they are currently typing in their note. Cards also group and organize this information to help clinicians gain insight about long-term trends and associations. Clinicians can additionally pin cards to the sidebar to create a persistent shared collection of information pertinent to the patient context.

4. We provide all these affordances with a passive and automatic design, which does not require active participation from the user.

In this way, we can simultaneously decrease documentation burden on physicians and use the captured clinical terms to aid physicians in information retrieval while typing a note. We elaborate on these features below.

9.4.1 Autocomplete

The backbone of the structured data capture within MedKnowts is a contextual autocomplete mechanism. We hypothesized autocomplete would enable structured clinical data capture without disrupting the existing documentation workflow and potentially even decrease keystroke burden on clinicians. Autocompleted terms could then be used to facilitate information retrieval and clinical decision support, offering longer term benefits.

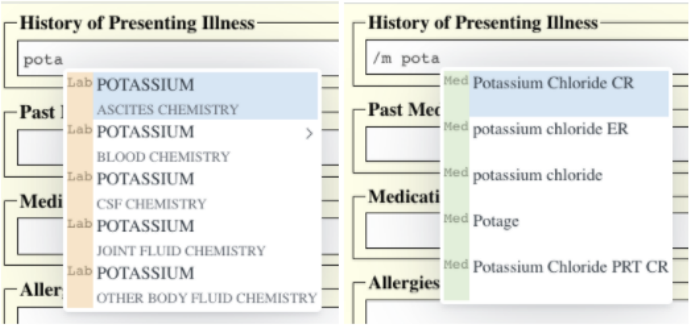


Figure 9-2: Autocomplete in the same context without filters and with filters. The "/m" command is used to limit the clinical terms displayed to medications.

We bootstrapped our autocomplete with a subset of clinical terms pulled from the SNOMED and UMLS medical ontologies [30, 194]. The ontologies contain abbreviations and synonyms for each term, allowing users to employ the language they are most comfortable with.

In our initial prototypes we use a single character trigger `/` to start the autocomplete, similar to *dotphrases* commonly found in EHRs [230]. When triggered, the autocomplete displayed a dropdown filtered to terms whose prefix matched the characters following the initial trigger. The clinician and scribes disliked the trigger because it required foresight that they were entering structured data or typing a recognizable concept and a priori knowledge of the set of recognized concepts. When no suitable term existed, users had to manually delete the trigger character.

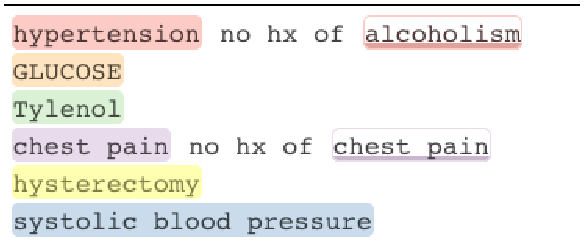
Therefore, our next iteration, outlined in Gopinath et al. [90], replaced the character trigger with a collection of rule-based triggers based on particular phrases, word boundaries, and punctuation. As an example within this paradigm, the phrase "presents with" is likely followed by a symptom, so the algorithm will show the autocomplete dropdown with symptoms listed first. User feedback indicated that rule based ranking is insufficient—the autocomplete often failed to display desired terms; and the boundary and punctuation triggers cause autocomplete to appear, unnecessarily, distracting the user.

To improve on the rule-based approach, we replaced the rules with a one-dimensional convolutional neural network model that predict when to trigger, and what type of clinical concept to prioritize, since a learned model can encode nuanced syntactical relationships. It significantly outperforms the rule-based triggering approach described in [90], achieving a precision of 43% versus 7%. Precision is defined as the fraction of times the user wanted to type a clinical concept when the autocomplete was triggered. In addition, after optimization, inference of this model requires an average autocomplete latency of about 18 milliseconds, which is close to the screen refresh rate and therefore perceived as instantaneous to the user.

While the model based approach works well, users indicated a desire to manually override the model—either forcing autocomplete to trigger or specifying the clinical concept to rank first. In these cases, we resort to slash filters: `/labs` or `/l` can be used to trigger an autocomplete context which is limited to labs. An empty slash forces autocomplete to trigger with the default ranking. An example of why filtering is useful can be seen in Figure 9-2. These filter shortcuts give users the fine-grained ability to easily insert structured information at any place in the note.

9.4.2 Post Recognitions

During prototyping users disliked that MedKnowts only identified clinical concepts entered with autocomplete. Unrecognized terms could appear because the user opted not to use autocomplete or because the user pasted text into the note. This issue was particularly noticeable when we used recognized terms to pre-populate later sections of the note. Some scribes would spend time re-entering unrecognized terms using autocomplete because they perceived the unrecognized term to be an error or wanted to generate the correct text later in the note. To resolve these issues, we implemented a version of the Aho-Corasick algorithm to automatically identify clinical terms from the text that has already been typed [9]. We dub this tagging mechanism *post recognition.*



hypertension no hx of alcoholism
GLUCOSE
Tylenol
chest pain no hx of chest pain
hysterectomy
systolic blood pressure

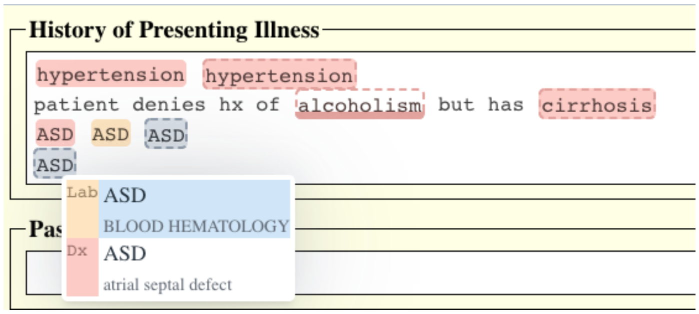
Figure 9-3: Autocomplete inserts terms as highlighted immutable chips. They can be deleted, moved around and copied like other text, but they cannot be modified.

9.4.3 Semantic Highlighting and Concept Disambiguation

As clinical jargon is notoriously overloaded, it is often the case that the same string can describe multiple terms [255]. For example, *Pt* can refer to a patient, physical therapy, or prothrombin time. While clinicians generally have the domain expertise to disambiguate between similar terms, jargon can create confusion for patients, medical trainees, and clinicians of a different specialty [283]. Therefore, MedKnowts needs to be able to correctly disambiguate each written term to its underlying clinical concept in the ontology in order for users to reap the benefits of contextual information retrieval features that our system offers.

MedKnowts uses live syntax highlighting to provide visual information scent about terms the system recognizes. MedKnowts supports six concept types: conditions, labs, medications, symptoms, procedures, and vital signs. When the user accepts an auto completion, the system inserts a *chip*—a highlighted block of text that can be copied, moved around, or deleted like other text. Each chip is highlighted with a color associated with its concept type; an example from each of the six concept types MedKnowts supports can be seen in Figure 9-3.

Post recognized phrases are also replaced with chips. However, while autocompleted phrases map to unique ontology items specified by the user's selection, post-recognized phrases can be ambiguous. In the case that a post recognition requires disambiguation, the user can click on the chip to select from the relevant set of candidate terms. Post recognitions are differentiated from autocomplete chips with a dotted border. When possible, the border also indicates the concept type with



History of Presenting Illness

hypertension hypertension
patient denies hx of alcoholism but has cirrhosis
ASD ASD ASD
ASD

Figure 9-4: Post recognitions are automatically recognized clinical terms. They are rendered with a dotted border and can be disambiguated through a popup menu on click. Negated post recognitions are rendered with an underline.

color: if multiple clinical terms match a post recognition but each clinical term is from the same concept type, the color for that concept type is applied to the entire post recognition. If clinical terms from multiple concept types match the post recognition then we display the recognition with a grey background. An example can be seen in figure 9-4.

Clinicians often reference clinical terms to indicate the absence of something, for example "no fever". In our initial prototyping we used double click to toggle chips between "positive" and "negated". When negated the chip is highlighted with an underline, and the text is transformed—for example "fever" becomes "no fever". Additionally, we provided autocompletions for each clinical term prefixed with "no" so that users could insert negated chips with autocomplete, but clinicians found this method of indicating negations brittle and disliked that lists of negated terms such as "no A, B, or C" had to be written as "no A, no B, no C" to comply with MedKnowts' simple negation implementation. To resolve these issues we implemented a modified version of negex [46] to automatically identify and highlight negated chips based on the surrounding text.

In the autocomplete dropdown ambiguity can arise when a string refers to multiple terms. For example, `potassium` refers to multiple labs measured with various fluids, so we display this disambiguating information as detail text in the dropdown, as seen in Figure 9-2.

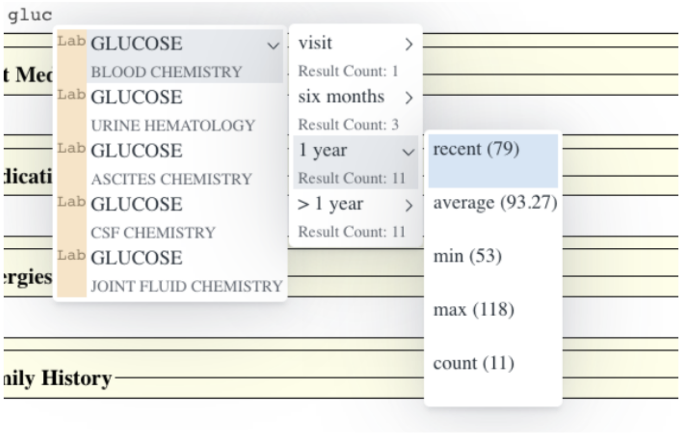
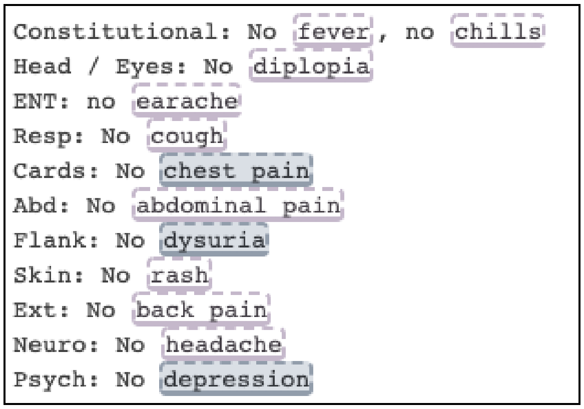


Figure 9-5: An example of context-specific information retrieval. Autocomplete insertion of lab results using a tree based menu with support for aggregation at multiple time frames and specific values

9.4.4 Context-specific information retrieval

To further aid clinicians, we automatically retrieve and display context-specific information from a patient's medical record. As an example, when a medication, procedure, or condition appears in the autocomplete dropdown, we use detail text–"in patient medical record" to indicate whether it previously appeared in the patient's medical record. We provide similar information scent next to chips with a small grey circle indicator.

This structured retrieval and display is particularly handy for documenting labs—after receiving requests to automatically insert quantitative lab results using autocomplete, we implemented a tree-based lab selection menu, displayed in figure 9-5. This hierarchical menu can be used to insert structured data associated with an autocomplete term. The user can select the name of the lab, a time frame based aggregate, or individual statistics within a time frame. The time frame aggregate is inserted as a string `LAB_NAME (MIN_VALUE - MAX_VALUE) AVG_VALUE` and individual statistics are inserted as a string `LAB_NAME STAT_NAME STAT_VALUE`. We also added the ability to insert vitals (pulse, heart rate, etc.) using the same methods, completing our set of clinical concept types.



```
Constitutional: No fever, no chills
Head / Eyes: No diplopia
ENT: no earache
Resp: No cough
Cards: No chest pain
Abd: No abdominal pain
Flank: No dysuria
Skin: No rash
Ext: No back pain
Neuro: No headache
Psych: No depression
```

Figure 9-6: An example of the review of systems section

9.4.5 Default Text

Medical notes are often pre-filled with boilerplate default text, but this text is often overwritten because it does not incorporate enough patient-specific context. MedKnowts further reduces data entry by taking advantage of structured data capture and using it to fill in later sections of the note. To this end, we created templates for each of the sections of the notes based on clinician input. When the user clicks on a blank note section the section is autopopulated with the template text, which is constructed using a mix of structured information parsed from the patient's medical record as well as clinical terms previously captured in the note. As an example, the Review of Systems (ROS) section (Fig. 9-6) is a boilerplate list of ten systems, and for each system the clinician has to describe the presence or lack of symptoms related to that system. MedKnowts automatically generates this ROS text for the clinician from text entered in previous sections—when a symptom is documented in the note, it is added to the appropriate line of the ROS template.

The addition of pre-populated text brought additional feedback from clinicians. Clinical terms are often associated with clarifying modifiers and specifiers and it is important to retain these modifiers and specifiers when copying forward clinical terms. For example "left lower abdominal pain" is more informative in diagnosing a condition than simply "abdominal pain". Expanding the ontology to include all possible combinations of modifiers for each term is not feasible. Instead we use a simple greedy algorithm to attach modifiers as prefixes to clinical concepts. This algorithm could be replaced with more advanced NLP methods but we chose to use this lightweight approximation to satisfy run-time requirements. The use of algorithms to detect negated and modified terms helps retain the nuance and meaning of the original text when copied across sections.

9.4.6 Concept-Oriented Views

Although there are multiple documentation systems in use at the hospital for writing ED notes, none of them are integrated with tools to view the patient's prior medical history. While some documentation systems provide limited views of a patient's information for the current hospital visit (e.g., recent labs or imaging), this does not help a clinician with reviewing and synthesizing the medical history. In order to access additional data, clinicians must still navigate through multiple different pages.

Some clinicians place two browser windows side by side and access data in one window and their note in another, others flip between pages and use their short-term memory to synthesize information. Both paradigms are error-prone—clinicians evaluate multiple patients in a shift and can easily navigate to the wrong patient's data or mis-remember details of patients with similar presentations. In addition, when interesting data such as a relevant note or lab trend is found by a clinician, there is no way to bookmark it for later use. All the computers in the hospital implement session timeouts to prevent the inadvertent sharing of patient information, so clinicians copy potentially relevant data into their note to preserve it and the surrounding context is lost.

MedKnowts reduces the need for clinicians to hunt for and retrieve data from multiple sources by proactively fetching relevant data and surfacing it just-in-time. To achieve this, we introduce the notion of a *card* for each clinical term in our ontology. Cards unify diverse information fragments related to the term in a single, templated, format. Each card has a header with the common name for the clinical term and synonyms for the clinical term from our ontology:

- Condition cards (e.g., `diabetes`)—display relevant medications from the patient's medical record, relevant vital signs, related procedures, and relevant snippets from notes in the patient's medical record.
- Labs and Vitals cards (e.g., `creatinine`, `blood pressure`)—display a box and whisker chart of lab values.
- Procedures and Medications cards (e.g., `hysterectomy`, `metformin`)—contain a list of relevant note snippets from the patient's medical history.

Note snippets are surfaced if they contained a mention of the term or a closely linked term and are ordered chronologically. The set of closely linked terms was algorithmically mined and a sample was validated by a clinician. Based on feedback, we excluded symptoms from our set of cards, as clinicians rarely needed medical history to contextualize symptoms.

9.4.7 Surfacing Cards

In our early prototypes we displayed cards in an attached sidebar when clinicians clicked on an associated chip within the note or another card. However this created a two step process to see any card—first type the term with autocomplete and then select the term to see the card. To reduce friction we automatically added a card to the sidebar for any term inserted with autocomplete. However autocomplete is a poor signal for whether a card is useful is in the long term. Cards added to the sidebar are displayed in a scrolling vertical stack. Cards can be removed, but left alone, they persist next to the note for the duration of the note authoring process, and useful cards can be pushed out of view as more cards are added. Some clinicians found this method of adding cards to the sidebar unintutive or confusing, and other clinicians felt like they were seeing too much irrelevant information. Additionally this method fails to surface post recognitions.

We eventually streamlined our approach to surfacing sidebar cards to a two-step process. Any time a term is recognized before the user's selection, we display the card for that term in a preview pane at the top of the sidebar. The preview pane displays one card at a time, and the card is not shared between users. Clinicians can pin a card displayed in the preview pane to move it to the sidebar. The cards pinned in the sidebar are persistent and are shared between multiple users. In this way, the sidebar becomes a collaborative record of the fragments from the patient's medical history that clinicians identify as being particularly important or relevant.

Cards are surfaced in the preview pane in one of three ways: first, they are automatically displayed when an autocompleted or post-recognized term appears before the user's selection; second, they are manually surfaced by users clicking on a chip within the note or another card; third, they are manually surfaced via a search bar at the top of the sidebar. Post recognitions with naming collisions (e.g., `pt`) must be disambiguated by the user before the associated card is surfaced.

9.4.8 Hand Designed vs. Automatically Generated Cards

Ideally, we could create individually designed and physician curated cards for all possible clinical concepts. But we do not have the resources to take that approach. Instead, during initial prototyping we created meta-cards for each clinical concept (labs, conditions) which act as templates for all clinical terms within that clinical concept. We describe the contents of the meta-card for each clinical concept in Section 9.4.6.

Automatically generated cards help solve a cold-start problem, as we hypothesized that users would be unlikely to adopt the system if the majority of clinical terms were associated with empty cards. But cards generated for a large number of clinical terms are slow to iterate on. For example, clinicians asked for certain labs to be added to cardiac cards. This type of change, if abstracted to all conditions, requires the development of a dataset to relate labs and conditions. While possible, finding or creating this type of dataset takes time. Conversely, adding lab values manually to cardiac cards is light weight and easy to validate with users. In the long term hand designed features could be replaced with generic models or datasets but in the short term we can iterate faster by taking a manual approach.

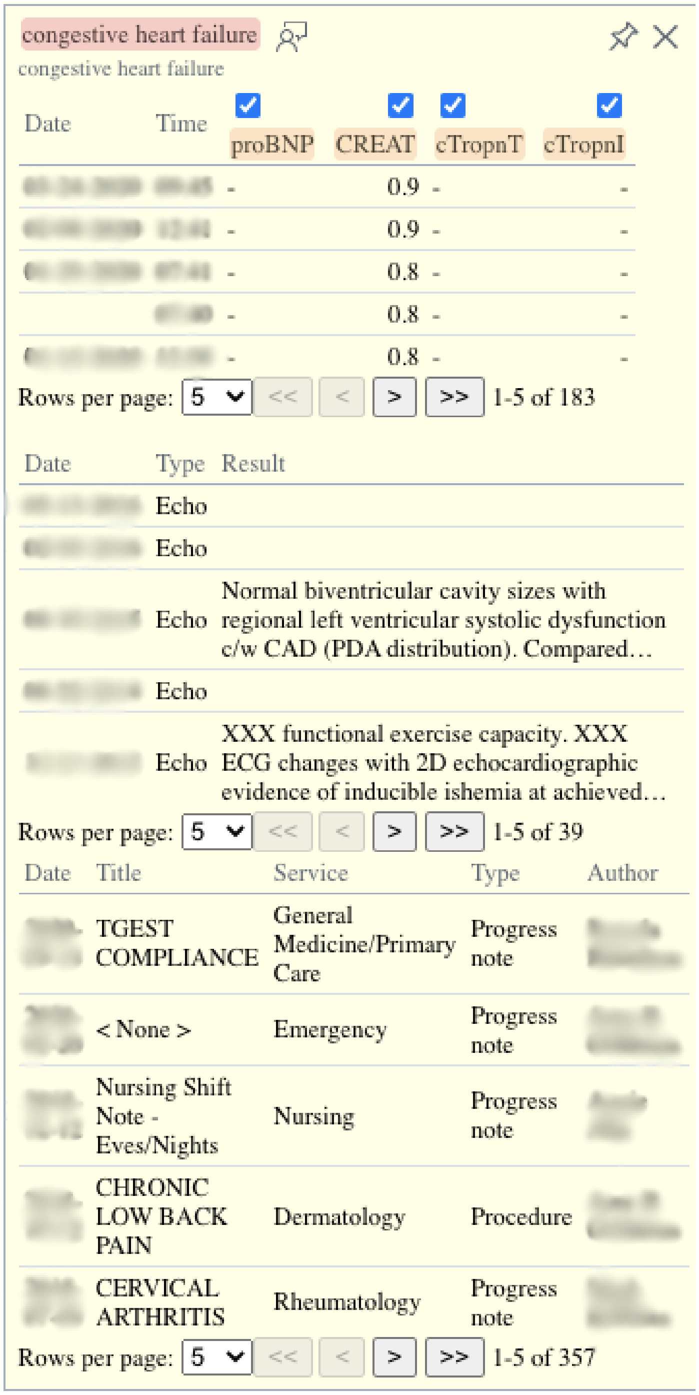
9.4.9 Card Design

Throughout the prototyping process clinicians consistently displayed strong preferences about the content of cards. We hypothesized that showing synonyms within cards would familiarize users with our ontology of terms. But clinicians found the inclusion of synonyms condescending because they *already knew that.* We received a similar response when we listed names of labs related to a condition on condition cards. However, clinicians reacted positively when we listed the names of labs along with their values since this is proactively fetching relevant information. Clinicians want to see information relevant to their decision making and other information is seen as noisy or unnecessary.

In addition, clinicians want information presented in the immediate format that they require; as an example, if the most recent lab value is the only useful piece of information, that is the only lab value that should be displayed. Conversely, some lab values can only be properly interpreted in the context of other lab values. For example, interpreting an elevated troponin values requires both prior troponin values and prior creatinine values. In that case both lab values must be displayed. We provided feedback forms on cards and accumulated various requests for data to be displayed on particular cards. However implementing granular changes for generic classes of cards is difficult..

To address this, in our second iteration of cards we chose to specifically focus on two types of cards: lab cards, and cards related to cardiac conditions. In the long run we expect that a set of a few thousand cards targeting individual clinical terms as well as general classes of clinical information (such as cardiac function) could support clinician's needs. While it is beyond our capacity to create an exhaustive set of cards, we can learn about and demonstrate the value of cards by creating a few for common terms. If proven valuable, other cards could be created by a small engineering team with clinical guidance, or even by clinicians themselves if given suitable authoring tools.

We worked in collaboration with three physicians to design a card which presents



congestive heart failure

congestive heart failure

Date	Time	☑ proBNP	☑ CREAT	☑ cTropnT	☑ cTropnI
[redacted]	[redacted]	-	0.9	-	-
[redacted]	[redacted]	-	0.9	-	-
[redacted]	[redacted]	-	0.8	-	-
	[redacted]	-	0.8	-	-
[redacted]	[redacted]	-	0.8	-	-

Rows per page: 5 << < > >> 1-5 of 183

Date	Type	Result
[redacted]	Echo	
[redacted]	Echo	
[redacted]	Echo	Normal biventricular cavity sizes with regional left ventricular systolic dysfunction c/w CAD (PDA distribution). Compared…
[redacted]	Echo	
[redacted]	Echo	XXX functional exercise capacity. XXX ECG changes with 2D echocardiographic evidence of inducible ishemia at achieved…

Rows per page: 5 << < > >> 1-5 of 39

Date	Title	Service	Type	Author
[redacted]	TGEST COMPLIANCE	General Medicine/Primary Care	Progress note	[redacted]
[redacted]	< None >	Emergency	Progress note	[redacted]
[redacted]	Nursing Shift Note - Eves/Nights	Nursing	Progress note	[redacted]
[redacted]	CHRONIC LOW BACK PAIN	Dermatology	Procedure	[redacted]
[redacted]	CERVICAL ARTHRITIS	Rheumatology	Progress note	[redacted]

Rows per page: 5 << < > >> 1-5 of 357

Figure 9-7: An example card surfaced for *Congestive Heart Failure*, which contains pertinent lab values, links to recent echocardiography reports, and recent notes that mention the condition.

information relevant to cardiac conditions. Our cardiac card includes labs and snippets from cardiac tests (EKG and Echocardiogram) and other free-text notes. An example of the cardiac card can be seen in Figure 9-7.

We augmented our lab card template to support multiple views of lab results. A table view can be used to see individual result values. When applicable, contextual labs that are useful for understanding the primary lab are added as columns to the table display. A zoomable line chart displays lab values over time and a box and whisker plot is used to display aggregate lab values over various time frames. Additionally we provided support for contextual lab results in the table view. For example, Kidney failure, which is measured by an elevated creatinine, leads to a build up of potassium, causing elevated potassium levels, a life-threatening condition that must be treated immediately. Whenever an abnormal potassium level is encountered, the next piece of information that is needed is what the kidney function is. We proactively provide this information by displaying creatinine levels directly on the potassium lab card.

9.4.10 Inline Display of Cards

Early on we realized it would be useful to access cards from within the note itself. We added the ability to hover on a chip to see a preview of the card.

9.5 Evaluation

As described previously, MedKnowts was deployed in two major iterations—one year of iterative prototyping and a one month evaluation. For approximately 7 months the prototypal deployments were used as the primary documentation tool by 1 physician (who is also a co-author) and 4 scribes across 1185 patients; the evaluation lasted 1 month and was used by the same physician and 4 scribes (2 scribes had participated in the prototypal deployments) across 234 patients. Our prototypal deployment ended after the hospital stopped using scribes in the wake of COVID-19; the second deployment began soon after scribes returned to the hospital. We could not do a comparative study against the baseline documentation system due to legal limitations

User	Patients	Shifts	Pins
P	150	12	58
S1	69	3	1
S2	50	4	0
S3	43	2	1
S4	33	3	15
Totals			75

Table 9.1: General usage data. Totals for Patients and Shifts are left out because the scribes worked with the physician on the same patient/shift. Pins reflect the number of cards each user pinned to the sidebar.

User	Autocomplete	Post Recognition
P	71	7
S1	35	0
S2	27	0
S3	6	0
S4	4	0
Totals	143	7

Table 9.2: How users inserted chips. Autocomplete indicates insertion via autocomplete and post recognition is disambiguation of post recognitions.

disallowing modifying the commercial note taking tool in use at the hospital, but we describe our evaluation below.

Prior to using the tool live in the ED, the scribes were introduced to the tool in thirty minute training sessions. In each session, the tool and its available features were explained to the scribe. After working in the ED the same co-author followed up with the scribes to get their feedback. At the time of the follow up the scribes each had used the tool for an average of 3 shifts (min 2, max 4) and completed an average of 46.5 notes (min 33, max 69). In the study follow-up, scribes filled out a system usability scale (SUS) [123] as seen in Figure 9-8, and answered questions from a script.

The final SUS scores were `[77.5, 77.5, 85, 95]` (avg. 83.75), the physician did not fill out a SUS scale. A score in the high 70s to upper 80s is considered to be good while a score above 90 is excellent [20]. These responses indicate that scribes found the tool relatively intuitive and useful enough to use frequently.

User	Search	Autocomplete	Post Recognition	Note Snippet
P	127	30	21	3
S1	4	4	25	0
S2	0	0	9	0
S3	0	1	7	0
S4	43	0	12	21
Totals	174	35	74	24

Table 9.3: How users added cards to the sidebar. Autocomplete means clicking on an autocomplete chip, post recognition means clicking on a post recognition, search means performing a search in the sidebar, and note snippet means clicking on a note snippet in a card. Clicking on a note snippet displays a card containing the full note text with the snippet highlighted.

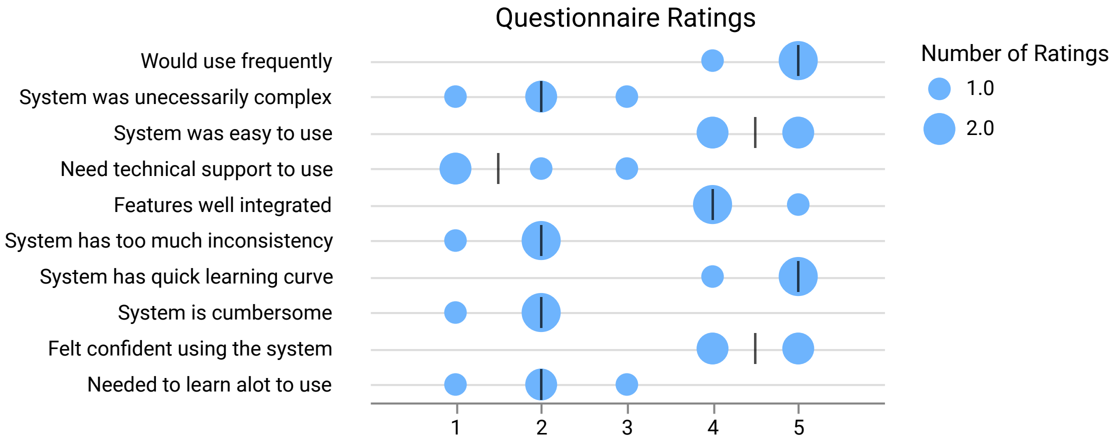


Figure 9-8: Scribes' System Usability Scale scores with medians displayed as black bars for each question

Feature usage, documented in Table 9.1, 9.2, and 9.3 as well as qualitative interviews yielded several takeaways. In both the tables and the rest of the evaluation we refer to the users as Physician (P) or Scribe 1-4 (S1-S4).

Most scribes described that autocomplete sped up their workflows but adoption of autocomplete changed based on scribe experience. S1, the least experienced scribe in the evaluation, noted that they liked autocomplete because they no longer needed to conduct internet searches to find correct spellings and obtain an understanding of the underlying concept space. S4, the most experienced scribe, found autocomplete less useful due to familiarity with terms, but still found utility for longer terms.

As users acclimated to the tool's functionality their usage changed. For example, S1 used autocomplete 8 times in their second shift and 27 times in their third shift. The increase in usage in the third shift was due to the use of autocomplete to insert lab values. It was unclear if the scribe had discovered this functionality on the third shift or had become familiar enough with the tool to adopt more advanced features.

Some of the differences in feature usage across scribes may be attributed to discoverability. For example S2 found the included references to other cards (card transclusion) to be very helpful, especially for getting more familiar with unknown terms, while S4 did not realize that they could hover on chips to see cards inline. S3 stated that card transclusion was helpful to quickly hover and get a sense of how central the concept is to the patient's history.

The lack of disambiguations for post recognitions may be due to the fact that post recognized chips both behave and look very similar to chips inserted with autocomplete. For example, if an ambiguous term is highlighted correctly and copied appropriately in default text, the scribe may not have any need to disambiguate it. Both S3 and S4 were happy that the system recognized terms but were unaware that post-recognized terms could be disambiguated.

Scribes appreciated the colored highlighting of embedded chips in the notes. They found that it allowed them to quickly scan what had occurred so far. For example, they could quickly skim through symptoms to orient themselves, and it was helpful that negated symptom mentions were visualized differently. One scribe mentioned that they could use the colors as an automatic visual aid to determine what components had been completed in the Medical Decision Making section, and what was left to be documented. This quick skim approach wasn't necessary for certain concept types (e.g., medications), but some scribes still found it useful for organization.

Scribes universally appreciated the default text that was auto-populated due to the structured data capture from autocomplete and post recognitions. This was most appreciated in Physical Exam and Review of Systems, despite imperfections in the default text. One scribe (S3) said it “made them much more efficient" allowing them to “get through charts faster." Another noted that the checkbox-based systems employed in the hospital's commercial EHR made it really easy to skip and miss an item, indicating the new system felt less error-prone due to its data entry.

At a high level, scribe experience correlated to the amount of synthesis of a patient's past history that was conducted, as advanced scribes had accrued more of the requisite clinical knowledge and reasoning and could handle documentation and synthesis simultaneously. S4 was the only scribe to examine past notes to try and find relevant information to share with the clinician. S4 liked the note snippets stating that "it saves me a lot of time compared to reviewing all of the patient's prior notes to simply be able to click...and have notes show up." In contrast S2 did not pay much attention to cards or read note snippets because they were focusing on documenting what the clinician was saying.

More experienced scribes were more likely to pin cards to the shared space. The more advanced scribe, S4 described their job as filtering information for clinicians based on relevancy and importance. Less advanced scribes perceived their role as recording rather than synthesizing or finding information.

Scribes integrated cards into documentation and retrieval processes for multiple concept types. For example, scribes mentioned using lab cards to compare a patient's current value to their baseline, clicking on a past procedure chip to pinpoint its date from surfaced notes, and leveraging condition cards to determine the extent and severity of a patient's existing condition. This indicates such cards act as information scent to guide scribes to important content. S4 described that they would use cards to dig into particularly relevant medical history; for example, if a patient with chest pain had past cardiac disease, the scribe would utilize the card to review "their previous work-up, notes from cardiology, and any prior surgeries/procedures." S3 noted particular utility in associated medications that were surfaced on condition cards; it prompted them to document, and the concept-oriented view also served an educational purpose of teaching them what was relevant, potentially aiding future synthesis.

While multiple scribes noted it was less useful for medications and symptoms, condition cards aided scribes in understanding the extent and trajectory of a patient's past condition. S4 noted that they "use it when... interested in more information about a patient's medical history, especially in a complex patient or a patient that is unable to provide a history due to acuity or altered mental status."

During the deployment MedKnowts proactively displayed 3614 cards to scribes as they typed, with a range of frequencies. For example, 53 hypertension cards were surfaced after typing "htn", while 144 terms were displayed only once each; these rarer terms included "spine fracture" and "lumbar spinal stenosis". Some short phrases that overlap with common language (e.g., `as` as `atrial stenosis`)were mistakenly tagged as clinical terms, but future iterations of MedKnowts can recognize these and omit them.

Experienced scribes tended to familiarize themselves with a patient's medical history prior to writing a note, and thus used the search bar to display cards prior to note-writing rather than triggering them automatically during the course of documentation. Future iterations of MedKnowts may want to support this workflow, since the existing information capture is focused on displaying information as the physician types rather than before the physician starts the note, which misses an opportunity to provide information scent before documentation begins. Less experienced scribes were more likely to click on chips within the note to see relevant cards. All scribes used chips to manually view cards in the sidebar, and all but one (S4) used chips as the primary tool for bringing up cards in the sidebar. The usage data reveals that users are willing to adopt a wide variety of techniques for accessing documentation, but appear to have significant preferences for one technique or the other.

9.6 Discussion

MedKnowts explores several interaction paradigms by enabling live automatic recognition of clinical terms within a medical note and displaying patient medical history in concept-oriented cards. Our iterative design process for MedKnowts underscored the need for EHR systems to embrace and augment, rather than replace existing clinical workflows. Our features were well received when they synergized with existing documentation practices. Implementing changes in a clinical environment is challenging, and clinicians and scribes are more receptive to such changes when presented with tools that are familiar and intuitive.

In future iterations of MedKnowts, we hope to expand on the possibilities enabled by fine grained linking of chips, in both the note and card interfaces, to standard medical ontologies. MedKnowts can leverage existing health knowledge graphs or outside resources that clinicians use, aiding their decision-making during documentation. Normalization to a standard ontology also allows notes to be translated to different audiences; medical acronyms can be automatically unravelled to layman's terms if a patient wants to understand their note. Clinicians with specific language preferences can also personalize note templates and autocomplete functionality with the vocabulary choices that they prefer.

Our observations from user interviews and interaction data have additionally presented new avenues for future work that could push forward these interaction paradigms. Clinicians often chunk information together. When a clinician wants to view a *Hemoglobin* lab, they are likely to search for *CBC* (Complete Blood Count), a set of laboratory tests, since *Hemoglobin* is recorded as part of a *CBC lab group*. MedKnowts could support such lab groups by adopting a wider clinical ontology, or even allowing clinicians to merge or combine cards within the user interface, dragging and dropping multiple lab cards together to create higher level lab groups. These modifications do not have to be limited to labs. Clinicians could, for example, group an insulin card with a diabetes card because the medication is directly treating the condition.

Providing clinicians with the ability to mold their information displays could not only help physicians synthesize medical records, but also create new possibilities for crowdsourcing rich labeled datasets of clinical relationships. Clinician-curated content would also be a potential solution for how to scale from a handful of manually curated cards to thousands of cards, and even create cards that serve different roles for different types of users (less granular for generalists like emergency physicians or primary care physicians, more granular for specialists like oncologists or immunologists).

In the opposite direction, clinicians sometimes want to refer to a specific value or event when recording information. When the clinician writes "patient has high glucose" it would be helpful if the system identified exactly which glucose lab was high, autocompleting not just glucose as a clinical term, but a specific measurement. By allowing clinicians to refer to granular as well as chunked information, we can get closer to the ideal of presenting information to the clinician in a way that mirrors their clinical thought process.

Cards could be improved by offering further custom views of the patient medical record, and providing context-dependent defaults. For example, the existing lab cards in MedKnowts display a table of result by default, but can display a line chart, and box and whisker plot as well. For some labs the most recent value is the only value that matters, and a table is appropriate, but for other labs, the trend over time is what matters, and a line chart would be more useful. A clinician's mental model of the patient becomes more refined over a patient's visit, resulting in different information needs as the visit progresses. An area of future work would be to investigate how to support this change. Contextual display of information in cards is challenging, but would continue to shift some of the cognitive burden of synthesizing the patient medical record onto the EHR.

We noted several occasions where usage differed based on clinical experience. However, even advanced clinicians can be novice users of the tool — experience level with MedKnowts is thus another dimension of the overall user experience. Ideally, a user interface would be intuitive for the novice user and provide support to help them grow into advanced users for the tool. Future work could expand on the logging we have implemented here to see how a user's usage of the tool changes over time, and what strategies we could employ to improve adoption of more advanced features.

There is a practical burden in scaling cards. However, because we use standard ontologies, we can leverage ongoing efforts to open-source physician-curated [244] and machine learned [182] concept maps. Users may benefit from the ability to manually author and customize default text templates and card contents but we hypothesize that a relatively small set of cards could cover most referenced terms. Semanik et al. [244] estimate that 150-200 concept maps could be used to cover the most commonly encountered conditions for a range of clinical specialties [244].

There is some risk that the automation provided by MedKnowts could lead to errors. For example a post recognition may be incorrect, modifiers may be skipped or added unnecessarily, and auto populated text could contain errors. Auto populated text requires manual verification, but this is an existing step in physician workflows, since the current system provides naive boilerplate text for modification. The risk of incorrect tags is lower than comparable clinical recognition systems [54, 237] because users are able to disambiguate terms and have information scent about the recognized terms.

There is additional risk that adopting MedKnowts could impede usability. To that end we have designed MedKnowts' features to be opt-in, leaving existing workflow unimpeded. Lastly we must consider if using MedKnowts could impact the responsible practice of medicine. The literature considers the risks of passive clinical decision support (CDS) like our concept-oriented cards to be minimal compared to active CDS [244]. While cards provide synthesized evidence, the responsibility is on the clinician to explore as needed, and data on cards links to the original source, e.g., note snippets expand to the full note.

9.7 Conclusion

MedKnowts captures structured clinical terms embedded within a free-text narrative and then links these terms to a concept-oriented, dynamic display of patient information that appears alongside a medical note. Thus, MedKnowts provides clinicians with a unified interface for writing a clinical note and exploring and navigating a patient's medical record. By integrating documentation and patient information MedKnowts lowers the cognitive burden of synthesizing the medical record, and demonstrates the possibilities of an EHR documentation system that can better serve clinicians.

We capture these clinical concepts via *autocomplete* and *post recognition*, and map them to standardized ontologies. This allows us to connect captured concepts with other medical databases and translate clinical terms for a variety of audiences. We provide patient information in a preview pane next to the note as the clinician types, proactively displaying contextual information when needed. A persistent sidebar of cards helps multiple clinicians develop a shared understanding of the patient and highlights important information. We prove the feasibility of our approach in a months-long deployment in an active ED, and demonstrate in our evaluation that clinicians are receptive to this approach.

Ultimately, we believe that MedKnowts has the potential to make clinical documentation truly work for clinicians by creating a live document that supports customized information retrieval, note-taking, and collaboration while simultaneously improving the final note that is shared with downstream doctors and patients.

Chapter 10

Conclusion

The data in electronic health records have immense potential to transform medicine both at the point-of-care and through retrospective research. However, structured data alone can only tell a fraction of patients' clinical narratives. As a result, a longstanding bottleneck has been the need to undergo time-consuming workflows to transform this data into the structured, usable form that is necessary to conduct robust, interpretable machine learning for health research.

This dissertation explored new methods and paradigms to improve the scalability of structuring data from clinical text. The work contained in the previous eight chapters has spanned the pipeline of clinical natural language processing: (i) schema definition, (ii) decision aid development, (iii) dataset creation, (iv) novel methods development, (v) retrospective evaluation, and (vi) live deployment and user studies. However, much work still remains to realize the goal of structured clinical data, both in translation of the existing work presented here and in new methods development. Further, it is important to acknowledge that many open problems in clinical information extraction are not machine learning problems, but rather sociotechnical ones, e.g., EHR interoperability.

Many long-standing goals in the clinical NLP community have recently become far more attainable due to the advances in the wider field. Over the course of the creation of this thesis (2018 to 2023), there have been massive paradigm shifts in natural language processing due to the introduction of (ever-larger) transformer models

[62, 35, 199]. These large transformer models have exhibited what have been called "emergent" behaviors, enabling them to conduct multiple steps of reasoning with far less data than before [293]. In the past months alone, several works have found that the current state of large language models can conduct medical reasoning, e.g., medical licensing exams [32, 253]. Further, there are new opportunities open in exploring novel workflows. Text is generated and ingested across all steps of clinical workflows: from clinical notes to patient communications, patient instructions, handoffs, and biomedical literature.

With the emergence of these models, we are at a crossroads for clinical NLP. The introduction of LLMs decreases our dependence on clinical annotation, which was often differentially available across institutions and diseases. This democratization of clinical information extraction could be particularly transformative for traditionally underfunded conditions. Additionally, LLMs have encountered a diverse set of clinical text in training data that has been generated by different stakeholders. As a result, they contain the ability to transfer to new settings. Moving forward, it will be important to invest in shared clinical LLMs, federated learning to enable learning on datasets across institutions, and the introduction of synthetically generated datasets that could enable more open science.

However, at the same time, these new capabilities are in tension with the cost of training and inference of large models: the financial cost, the environmental cost, and the opportunity cost of other NLP research. It will become important to distinguish when large models may be necessary versus when they merely are a one-time source of weak labels. Further while text is ubiquitous in clinical workflows, it is only one lens into patient state. Thoughtful understanding of existing workflows and development of new benchmarks will be required to enable judicious application of large language models that lead to clinical impact.

Despite the rapid changes in language models, the core ideas within this thesis remain relevant. As models become more powerful, approaching AI from a human-centered perspective only becomes more urgent. It will become imperative to design systems that jointly optimize for the strengths of humans and machine learning (and understand failure models of this interaction). Further, as language models grow larger, it becomes pertinent to interact with them in a grounded and principled manner, to avoid hallucinations. Finally, AI-assisted writing and information retrieval are increasingly feasible, making it important to understand the engineering and integration considerations when modifying clinical documentation workflows.

Together with the increasing availability of data from new modalities (e.g., genomics and wearables), we are at an inflection point for personalized medicine. Below I detail a few directions for future work that could leverage clinical text to improve the experiences of clinicians, researchers, and patients alike.

Clinical Researcher-driven Information Extraction

Structuring EHR data is central to many clinical research studies. While this dissertation has proposed many methods to decrease the effort required, these methods require a data scientist to iterate and deliver models, which can be slow and costly. Given recent advances in language modeling, less technical expertise and labeled data could be required to jumpstart the extraction of structured data, allowing researchers to query and explore their own data. This direction requires the development of clinical language models that can be used locally in the researcher's environment, for data privacy, and methods for updating these models that can account for changing clinical practice (e.g., new medications).

However, ultimately, LLMs have been known to be sensitive to exact wording of prompts. Given the high-stakes nature of healthcare, it is imperative however that clinician's intentions match up with the model's interpretation, and that this is an easy. Therefore, such research will require a human-centered focus to design systems around how clinicians think and manifest their domain expertise. Given the current high cost of chart review, I believe clinician-driven data science can catalyze the rate of retrospective research, by allowing clinicians to more easily explore the patterns in their data, engage in faster hypothesis generation, and augment their datasets with discovered confounders.

Improved EHR Interfaces

MedKnowts has only scratched the surface of how we can improve the clinician experience in EHRs. Our existing system retrieves the relevant data corresponding to specific clinical concepts that were typed in or queried; for example, if a clinician were to autocomplete “diabetes,” we would automatically surface diabetes-pertinent information, e.g. A1C lab trends, recent endocrinology notes. However, often times, information needs are more fluid and require a greater degree of flexibility in the forms of information retrieval. Clinicians would ideally be able to ask free-form questions of the patient record (e.g., “Has the patient had any side effects with drug X?”). Medical question answering from the patient record has been of long-standing interest, with the release of several datasets [149, 200, 318]. Given the improvements in large language models to instruction-based interfaces [199], it is a ripe time to explore this area further. Particular modeling complexity comes from the fact that a patient’s disease state is constantly evolving, so it is important to understand how to correctly conduct information retrieval on a timeline of ordered documents, rather than a set.

An additional direction of note is AI-assisted note writing, particularly as it pertains to retrieved material. If implemented, such changes could enable physicians to spend less time with screens and more time with patients, improving patient satisfaction and combating physician burnout. Current research in this area has largely focused on AI-assisted writing that utilizes the clinician-patient conversation for source data [135, 240], but a huge amount of time also goes into synthesizing the patient’s past history, as described in Chapter 9. Technical challenges arise from the high-stakes nature of healthcare documentation.

A failure mode of current language models is their occasional tendency to *hallucinate* facts and provide answers for which there may be no evidence. Much of the existing literature on AI-assisted writing has focused on creative writing tasks. As with other technical writing, clinical documentation has additional requirements around veracity and requires conditioning on specific patient contexts. Building AI-assisted writing into EHRs will require new NLP methods and corresponding interfaces with the appropriate guardrails.

Particularly relevant here are the findings from Chapter 4, in which we found that AI-assisted decision aid can decrease a clinician's agency. While the autocomplete functionality in Chapter 8 remained relatively limited in its scope, integration of the functionality to autocomplete longer strings requires more careful thought [106]. For example, researchers recently found that users with AI assistants wrote less secure code [206]. As clinical documentation is often used as a scratchpad for medical reasoning and decision-making, it is imperative to strike the balance between assistance to prevent clinician burnout and assistance that biases the end user.

Translation to Patient-friendly Language

While this thesis focused on use cases of text for providers and researchers, clinical notes can also be valuable for patients. Unfortunately, clinical text can already be difficult even for clinicians of other specialties to read, not to mention patients. In a study of discharge summaries, only a quarter of patients could adequately understand their surgical summary, and 65% didn't have the reading level required [51]. Additionally, there is great opportunity in increasing patient comprehension of their medical data: approximately a fifth of patients find errors in their own notes, when prompted [25], and approximately a third of patients forget prescription information described to them during appointments [269].

Translating notes to a patient-friendly language accessible to a wider population could therefore be useful for reducing EHR errors and increasing patient adherence and understanding. Existing efforts in the clinical domain have struggled on overloaded terms, where translation would be the most helpful [296]. General translation methods (e.g., English to French) are not directly applicable, since they often assume you have access to text samples from the target language distribution. However, there do not exist corpora of clinical notes in layperson language, and therefore one must draw signal from other sources. Our LLM results on clinical disambiguation in Chapter 7 provide a proof-of-concept that such models may now be sufficiently powerful to generate approximate translations, making this task ripe for revisitation.

Appendix A

Additional Information for Chapter 2

A.1 Description of Annotation Tool Suggestions

Below, we describe the lightweight suggestion system that underlies our annotation tool. Due to its simplicity, it can run seamlessly. First, we constructed two data structures from UMLS tables, using only the terms from SNOMED and RxNorm.

- A lookup table. This is a dictionary mapping text to all of the CUIs that have that text as a Preferred Name or Synonym.
- An inverted index. The inverted index was built as per classical information retrieval. The 'document' for a CUI was a concatenation of stemmed versions of all the words in that CUI's Preferred Name or Synonym.

The suggestions then consist of:

- Any direct matches of the text span, found in the lookup. They are sorted by the number of synonyms they have, which we use as a proxy for population prevalence.
- The closest set of matches from the inverted index. Before searching, we remove stop words from the highlighted text and stem the remainder of the words. Among equally relevant matches, we again sort by number of synonyms as a proxy for population prevalence.

A.2 Performance of Systems by Semantic Type

Below we show the performance of the top 10 systems of the n2c2 Shared Task on the test set, subdivided by the 20 most common Semantic Types. We note that a CUI can have multiple Semantic Types, in which case it is included in several calculations.

Test Subset	**# of Examples**	**Max Accuracy**	**Avg Accuracy**	**Pooled Accuracy**
Disease or Syndrome	713	88.8%	85.7%	96.2%
Finding	674	78.8%	75.1%	86.4%
Pharmacologic Substance	667	93.9%	91.9%	97.2%
Laboratory Procedure	665	88.7%	85.8%	94.4%
Organic Chemical	624	95.5%	94.4%	98.1%
Therapeutic or Preventative Procedure	565	77.5%	72.7%	89.4%
Sign or Symptom	456	94.3%	92.3%	96.9%
Diagnostic Procedure	411	90.3%	84.6%	95.4%
Qualitative Concept	311	94.9%	89.0%	96.8%
Health Care Activity	299	84.6%	81.6%	89.3%
Spatial Concept	241	92.5%	89.0%	95.4%
Pathologic Function	228	89.0%	85.8%	94.7%
Body Part, Organ, or Organ Component	189	82.0%	65.2%	89.9%
Quantitative Concept	165	95.8%	92.8%	97.0%
Temporal Concept	122	85.2%	83.6%	91.8%
Medical Device	108	78.7%	70.0%	88.0%
Antibiotic	102	97.1%	95.6%	98.0%
Neoplastic Process	100	91.0%	84.8%	97.0%
Functional Concept	93	87.1%	77.0%	88.2%
Amino Acid, Peptide, or Protein	71	91.5%	86.9%	91.5%

A.3 Full Clinical Annotation Schema Guidelines

Overall Guidelines

Tag all words that seem like medical terms, i.e., they require medical knowledge to understand or are frequently used in medicine.

Concepts and Actions (Nouns and Verbs)

- Tag all labs, e.g., "WBC count", "hematocrit", or "urinalysis".
- Tag all procedures, e.g., "EKG", "chest x-ray", "fingerstick", or "CT".
- Tag all treatments or interventions, e.g., "transfusion", "aspirin", "CABG", or "surgery".
  - Tag interventions even if they're common non-medical concepts, e.g., "fluids" or "tube".
  - Tag normal English verbs that are used to describe an intervention, e.g., for “blood cultures were drawn” tag “drawn” as “sample obtained”.
  - For drugs, tag the drug name and dosage route, but do not tag the dosage. e.g., do tag “PO” but do not tag “15mg”.
- Tag all symptoms, e.g., "headache" or "nausea".
  - Tag symptoms even if they are not medical terms, i.e., for “chest pain” tag “chest”, “pain”, and “chest pain”.
- Tag all findings, e.g., "respiratory distress", "edema", "blood pressure", or "heart rate".
- Tag all diagnoses, e.g., "COPD", "MI", or "osteoporosis".
- Tag all medical history elements, e.g., "trauma" or "smoking history".

- Tag all healthcare terms, e.g., "PCP", "ED", "ambulance", "EMR", "OSH", "cardiologist", "rheum" (or any other medical specialty), "follow-up", "review of systems", "past medical history", "patient", "assessment", or "consult".
- Tag all illnesses or medical findings, e.g., "injury".
- Do not tag negation words, e.g., for “denies chest pain” do not tag “denies”.

Modifiers (Adjectives and Adverbs)

- Tag all words that modify a medical concept.
- Tag all medical modifiers, e.g., "acute", "pleuritic", or "friable".
- Tag all anatomical terms, any words that refer to body positioning, e.g., "orbital", "right", or for “chest x-ray” tag “chest”.
- Tag all descriptions of findings or a patient’s state, e.g., "febrile" or "stable".
  - Do not tag adjectives that describe something medical but do not have a separate medical meaning from their English meaning, e.g., for “vital signs normal” do not tag “normal”.
- Tag all descriptions of diseases or diagnoses that are considered medical terms or have separate medical meanings, e.g., "acute", "chronic", "mild", "moderate", "severe", "dynamic".
  - For example, tag “sensitive” if it is in the context of antimicrobial sensitivity, e.g., for “blood cultures grew E. Coli sensitive to x, y, z” tag “sensitive” because it describes a medical concept of antimicrobial sensitivity.
  - Do not tag a term if you search for it and there’s no tag available, e.g., "profuse".
  - Do not tag the words "likely", "unlikely", or "possible".

References to Parts of the Note

- Tag references to parts of notes even though they are English words, but do not tag individual words that comprise them.
    - For example, "HPI" (History of Present Illness), "PMH" (Past Medical History),"A/P" (Assessment and Plan)

Span Splitting

- Tag sub-concepts if they have their own medical meaning, e.g., for “extraocular movements intact” tag “extraocular”, “extraocular movements”, and “extraocular movements intact”. However, in the case of "respiratory failure" the word "failure" does not have any separate medical, anatomical, or physiological relevance so it would not be tagged.
- Do not tag sub-concepts if they are contained within other concepts that express the same meaning more fully written out, e.g., for “Crohn’s disease” tag “Crohn’s disease” and “disease” but not “Crohn’s”.
- Do not tag sub-spans of very common administrative terms used to describe portions of a note, e.g., for “history of present illness” do not tag “history”, “present”, or “illness”.

Multiple Concept Matches

- Tag multiple CUIs when there are multiple that appear to be a good match in context, e.g., for “fever” potentially tag “fever symptom (finding)” and “fever (sign/symptom)”.
- Even if a term can have multiple taggable meanings, only tag the meaning in context.
- If two terms match equally well, only tag the verbatim text, e.g., for “acute kidney injury” only use tag “acute kidney injury” but not “kidney failure, acute”.

No Concept Matches

- If there is no exact match in SNOMED, tag as “CUI-less” (and optionally any close approximations)
    - If both subconcepts in a compound concept are tagged, there is no need to tag the compound concept as "CUI-less."
    - Do not use “CUI-less” to tag terms that are only semantically off (the concept "patients" for "patient").
- If there is an approximate match, feel free to tag as both "CUI-less" and those close approximations, e.g., "presentation" for "presents with".

Special Considerations

- If a term repeats within the same note, tag the term every time it appears.
- Ignore typos and tag the term as its intended meaning.
- Tag common medical abbreviations, e.g., "h/o", "PMH", "ggt", or "p/w".
    - Do not split abbreviations into different spans, even if they represent different medical concepts, e.g., for “EOMI” do not tag “EO” and “EOM” separately.
- For simplicity, tag lab tests with priority order level (aka finding) > measurement (procedure) > substance, e.g., for “albumin level” tag with “albumin level measurement” instead of “albumin”. While multiple may be equally valid, this can be fixed in post-processing.
    - If a phrase describes the results of a lab test, only tag the substance tested itself and not the modifier, e.g., for “low albumin” tag “albumin” and do not tag “low”.
    - Similarly, do not tag the numeric value if given, e.g., for “INR 3.1” do not tag “3.1”.

  - Do tag lab test results with modifiers if the resulting state has a name, e.g., for “elevated WBC count” tag the phrase as “leukocytosis”.

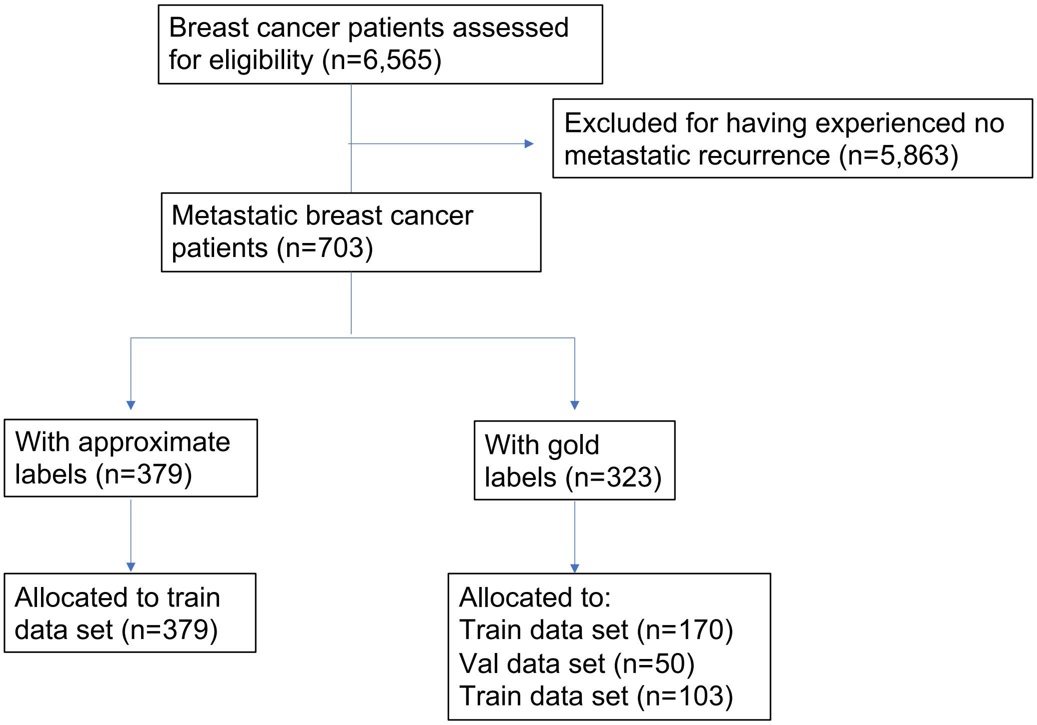
- Do not tag units of measurement.

Appendix B

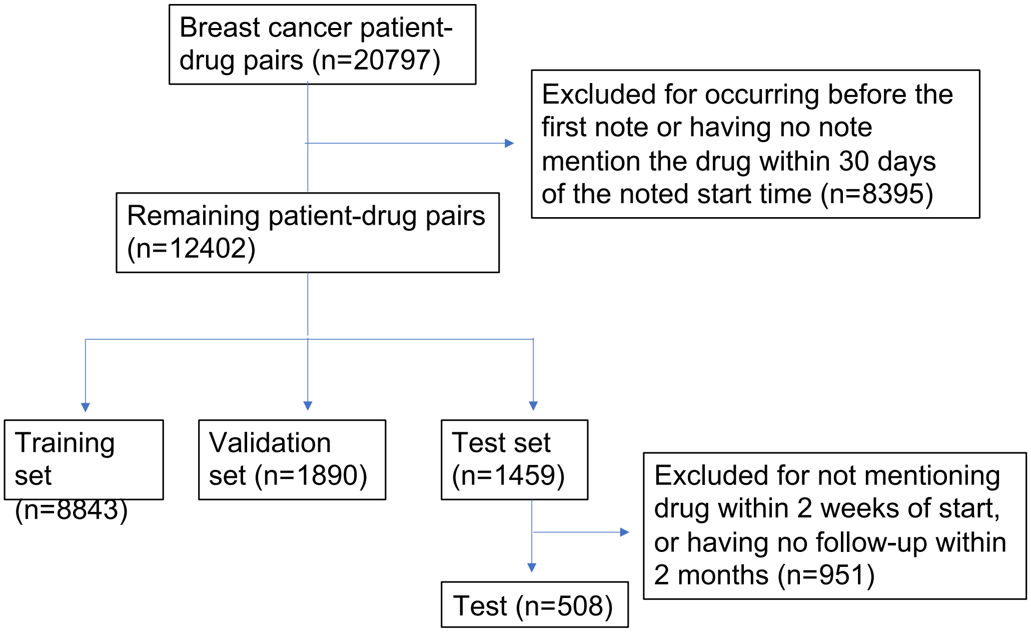
Additional Information for Chapter 5

B.1 Consort Diagrams

Below is the consort diagram for the metastatic extraction task.


Breast cancer patients assessed for eligibility (n=6,565)
Excluded for having experienced no metastatic recurrence (n=5,863)
Metastatic breast cancer patients (n=703)
With approximate labels (n=379)
With gold labels (n=323)
Allocated to train data set (n=379)
Allocated to:
Train data set (n=170)
Val data set (n=50)
Train data set (n=103)

Below is the consort diagram for the oral therapy extraction task.


Breast cancer patient-drug pairs (n=20797)
Excluded for occurring before the first note or having no note mention the drug within 30 days of the noted start time (n=8395)
Remaining patient-drug pairs (n=12402)
Training set (n=8843)
Validation set (n=1890)
Test set (n=1459)
Excluded for not mentioning drug within 2 weeks of start, or having no follow-up within 2 months (n=951)
Test (n=508)

B.2 Proof of Reduction to Binary Search

Theorem. *The Model-derived Query Utility formulation collapses down to binary search in a zero-information scenario (uniform time and probability distribution).*

Proof. Without a loss of generality, we can say that the endpoints for any given iteration are 0 and t, and that T[j]=j. By definition of the uniform distribution, the marginal probability at any given point j can be calculated by $p(y^j = 1|0, t) = j/t$. Then, since $p(y^{\frac{t}{2}} = 1|0, t) = 1/2$, we have that $\hat{y} = \frac{t}{2}$. Given our $\hat{y}$, we do casework to find the j that corresponds to $\arg\max_j E[\Delta^j]$. We let $\hat{z}$ be the latent labeling corresponding to $\hat{y}$.

Case 1: $\hat{z}^j = 1$

The case that $\hat{z}^j = 1$ occurs with probability j/t, making $(a^*, b^*) = (0, j)$ and $\hat{y}_[a^*, b^*] = j/2$.

Case 2: $\hat{z}^j = 0$

The case that $\hat{z}^j = 0$ occurs with probability $1 - j/t$, making $(a^*, b^*) = (j, t)$ and $\hat{y}_[a^*, b^*] = (t - j)/2$.

Plugging in, $E[\Delta^j] = (j/t)(t/2 - j/2) + (1 - j/t)(j/2)$. This quadratic has its maximum at $j = t/2$, the halfway point of the interval. This indicates that the point that maximizes *Model-derived Query Utility* over a uniform distribution is the midpoint, equivalent to binary search. □

B.3 Reinforcement Learning Details

B.3.1 Decision-making agent

We view this formulation as a Markov Decision Process, where an agent views a state $s \in \mathcal{S}$ that encodes information about the past queries Q and the representations of the extraction model, f, conditional on the queries. The agent must choose an action $a \in \mathcal{A}$ that corresponds to either querying an index i, or making a final extraction.

The process terminates when the agent makes a prediction, or when the agent can verifiably determine the true labelling of the sequence. After termination, a reward is determined by the final accuracy of the extraction model, conditioned on the queried information.

State

At every state, we retain a history of past queries $Q = \{q_1, \cdots, q_m\}$. Our state s_i is composed of three components: a positional encoding of every index, the predicted marginals $p_\theta(y_i|\{q_1, \cdots, q_m\})$, and the final-layer hidden states of the prediction network at every index, h_i.

Positional Encoding Following Vaswani et al. [281], we use a sinusoidal positional encoding. Formally, for every index i, we define our positional embedding vector $\overrightarrow{p_i}$ of length 64 as

$$\overrightarrow{p_i}^{(j)} = \begin{cases} \sin(r^{\frac{j}{64}} \cdot i), \text{for } j \text{ is even} \\ \cos(r^{\frac{j-1}{32}} \cdot i), \text{for } j \text{ is odd} \end{cases}$$

where j represents the indices of $\overrightarrow{p_i}$, spanning from 0 to 63.

Predicted Marginals Using the predicted probabilities of our model $p_\theta(y = i|X, Q)$ at every index i, it is possible to induce marginals

$$\hat{z}_i = p_\theta(z_i = 1|X, Q) = p_\theta(i \geq y|X, Q) = \sum_{i \geq y} p_\theta(y = i|X, Q)$$

Hidden States The third component of our state is the hidden states of the extraction model at every index, h_i. We concatenate all components to form our state, $s_i = [p_i, \hat{z}_i, 1 - \hat{z}_i, h_i]$.

Action

Our action space is $\mathcal{A} = \{q_1, \cdots, q_n, p\}$, where $a = q_i$ indicates the act of querying the oracle for the label of note i, and $a = p$ indicates the act of making a final prediction.

Reward

As training signal for the policy model, we may use a scalar reward which represents how well our extraction model performed after using all query information. However, using a delayed reward at the end of each example makes learning difficult. Instead, we advocate for reward shaping [192], where intermediate rewards are provided to accelerate the learning process. Thus, we define the reward at a given state as

$$R(s_{i-1}, a) = \begin{cases} Acc(y, f_\theta(X|q_1, \cdots, q_i)) + Acc(y, f_\theta(X|q_1, \cdots, q_{i-1})) - C \text{ if } a \neq p \\ 0 \text{ if } a = p \end{cases}$$

where C is a hyperparameter for the cost of querying. In other words, the reward is the incremental improvement of querying, offset by a query cost. When the query cost outweighs improvement in accuracy, the model is incentivized to quit the process by making a prediction. There are many possibilities for parametrizing $Acc(y, \hat{y})$, but here we use $Acc(y, \hat{y}) = |y - \hat{y}|$.

B.3.2 Reinforcement learning

We use a reinforcement learning approach to learn a good policy for our agent. Formally, we define a policy network $\pi_\beta(s) = p_\beta(a|s), s \in \mathcal{S}, a \in \mathcal{A}$ that assigns probabilities to actions, given the current state of the agent.

We aim to find a set of values for β that maximizes the expected reward under the policy π_β. Thus, our objective is to maximize

$$J(\beta) = E_{(s_1,a_1,s_2,a_2,\cdots)}[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t)]$$

where actions $a_i \sim \pi_\beta(s_i)$ are sampled from the policy and the next state $s_{i+1} \sim p(\cdot|s_i, a_i)$ is obtained via the MDP transition function. In this setting, γ is the discount factor of the MDP.

We optimize $J(\beta)$ by applying the policy gradient theorem and the REINFORCE algorithm [301]. First, we sample a trajectory under the current policy π_β to obtain $\{s_1, a_1, s_2, a_2, \cdots, s_T, a_T\}$, where $a_i \sim \pi_\beta(s_i)$. In order to compute an estimate of the gradient of our objective, we first compute the cumulative reward at every step, $v_t = \sum_{j=t}^{T} \gamma^{t-j} R_j$. Our gradient then is

$$\nabla_\beta J(\beta) = \sum_{t=0}^{T} v_t \nabla_\beta \ln \pi_\beta(a_t|s_t)$$

Policy Architecture

We parametrize our policy model using a 1-layer Transformer encoder block as in [281], followed by a fully-connected layer. Because there are $n + 1$ actions for an input of length n, we concatenate a trainable bias to the output of the fully-connected layer before passing it through a softmax function. In our experiments, we use a transformer with a hidden dimension of 64 and 16 attention heads.

Training Scheme

We train for 20 epochs using batched gradient ascent over sampled trajectories. In order to facilitate training, we use a warm-up scheme that linearly increases the hyperparameter C from 0 to its desired value every epoch. We train using the Adam optimizer with initial learning rate 1e-3. All hyperparameters were selected using cross-validation over the validation set.

Appendix C

Additional Information for Chapter 6

C.1 Synthetic Experiments

C.1.1 Synthetic Distributions

In this section we describe the distributions for the synthetic experiments in Section 6.5.1. Both of these distributions are members of the class of distributions from Section 6.4.1. We proved in the full manuscript [6] that all of these distributions satisfy Assumptions 1-3, and that S is the optimal representation for OCP on these distributions.

Distribution 1 (Figure 2a)

Our first synthetic distribution includes trajectories X of length 10 with $d = 8$ features that are generated as described below:

- The set S consists of 4 time-irreversible features. Each feature in S has a fixed probability of activating (switching from ‘0’ to ‘1’) over the entire trajectory, independent of the other features in S. Activation time was chosen uniformly over the whole trajectory and independently per feature, and once activated, features remained on. In our synthetic data, probabilities were 0.4 for the first two features and 0.6 for the next two features.

- The set $\hat{S}$ consisted of noisy versions of the first three variables in S. ϵ_i was set to 0.7 for all variables.

- The last feature X_{-1} was a background, reversible feature in B. It was a periodic function, alternating between 0 and 1, with uniform initialization over $\{0, 1\}$.

Distribution 2 (Figure 2b)

Similarly, our second synthetic distribution includes trajectories X of length 10 with $d = 7$ features that are generated as described below:

- The set S consists of 4 time-irreversible features, identical to before. Each feature in S has a fixed probability of activating (switching from '0' to '1') over the entire trajectory, independent of the other features in S. Activation time was chosen uniformly over the whole trajectory and independently per feature, and once activated, features remained on. In our synthetic data, probabilities were 0.4 for the first two features and 0.6 for the next two features.

- The set $\hat{S}$ consisted of noisy versions of the first two variables in S. ϵ_i was set to 0.55 for all variables.

- The last feature X_{-1} was a background, reversible feature in B. X^0_{-1} was sampled uniformly from $\{0, 1\}$ and the rest of X_{-1} was set such that for all $t \geq 1$, $\mathbb{P}[X^t_{-1} = X^{t-1}_{-1}] = 0.3$.

C.1.2 Experimental Setup

Synthetic Data Creation

In order to quantify how many unlabeled pre-training samples are required to recover S, we created data sets $\{X_i\}_{i=1}^m$ with varying m, taking on values 50, 100, 200, 400, 600, 800, 1000, 2000, 4000, 8000, and 16,000. For each dataset size m, we generated 100 independently drawn sets, according to the distributions described previously.

For each created set, a single pair was sampled from each trajectory according to OCP, PCL, or OCP-biased sampling. I.e., an unlabeled dataset of m trajectories produced a pre-training dataset with m pairs.

Selection of Optimal Representation

Now, for each of our pre-training datasets and each sampling scheme (OCP, PCL, OCP-biased), we now determine what feature representation $\hat{g}$ would be selected via pre-training for each one.

In order to do so, we iterate over each possible $g \in \mathcal{G}$ where $\mathcal{G} = \{U \subset [d] : |U| = |S| = d_0\}$. After subselecting to the features in g, each pair of data points $(X^t, X^{t'})$ was featurized as input as $[X^t; X^{t'}; X^t - X^{t'}; |X^t - X^{t'}|]$. The loss is then minimized via the LogisticRegression implementation from scikit-learn with the ‘liblinear’ optimizer. We then select the $\hat{g}$ that minimizes the empirical pre-training loss and calculate the overlap with the true features in S.

C.1.3 Explanation of Observed Behavior

Distribution 1

In Distribution 1, we see that OCP essentially always converges to the optimal representation with 8,000 data points, and OCP-biased also coverges, albeit slower. However, PCL is never able to break the barrier of 3 features, since it opts to choose the periodic feature X_{-1} instead. This feature (which alternates between 0 and 1) is highly discriminative for the PCL pre-training task, since $|X_{-1}^t - X_{-1}^{t+1}| = 1$ whenever $Y = 1$, and the same is true only half the time when $Y = 0$. Therefore, while it is a background, reversible feature that may not be useful for a downstream task, it is useful for the PCL task, and therefore PCL fails to find the optimal representation of time-irreversible features. In contrast, for OCP, $|X_{-1}^t - X_{-1}^{t+1}| = 1$ is true across all examples, and therefore would not be chosen as a discriminative feature.

Distribution 2

In Distribution 2, we again see that all methods are able to identify the optimal representation. However, OCP requires fewer pre-training samples in order to reach the optimal representation. While not periodic as in Distribution 1, X_{-1} is again weakly predictive of whether or not a window is consecutive. While X_{-1} is not as strongly predictive as in Distribution 1, PCL still opts to select it in $\hat{g}$ in the lower-data regime. However, with sufficient data, PCL overcomes the “false” signal to opt for the “correct” feature instead.

C.2 Real-world Experiments

C.2.1 Pre-training for feature selection

Implementation Details

For the linear pre-training for feature selection experiment, we utilized the LogisticRegression implementation from scikit-learn with the 'liblinear' optimizer and balanced class reweighting [205]. Hyperparameter tuning was conducted on the validation set independently for each feature subset, dataset size, and fold number. Hyperparameters included regularization scheme ('l1' or 'l2') and regularization constant (10^{-3} to 10^{6}). All performance reported is on the held-out test sets using the best hyperparameter setting from the validation set.

Granular Experimental Results

Below, we present a more granular view on results including standard deviations, focusing on the difference between direct downstream prediction and using order contrastive pre-training to select features.

	Fraction of training data				
Features	**1**	**1/2**	**1/4**	**1/8**	**1/16**
OCP subset	0.864 ± .022	0.860 ± .026	0.847 ± .024	0.808 ± .023	0.786 ± .068
All features	0.856 ± .021	0.851 ± .021	0.818 ± .036	0.723 ± .057	0.726 ± .074
OCP - All	0.008 ± .014	0.008 ± .039	0.029 ± .035	0.082 ± .058	0.054 ± .050

Table C.1: AUC ± standard deviations (as calculated over five folds), comparing the use of all features to just those selected by OCP. The final row displays the difference in AUC between OCP-selected features and all features and the standard deviation of that difference.

Average Precision Results

In addition to the note-level AUC, we also provide results via a different patient-level precision metric; this provides another interpretable view on performance, since each patient has multiple notes in the test set, and the labels are imbalanced. We define *Average Precision at 80% recall* in the following manner. We first find the threshold at which 80% of positive labels (displays progression) would be recovered. Then per patient, we calculate the precision of the retrieved notes, assuming use of that threshold, which we then average over patients. If no notes are surfaced for a patient, we set the precision to 1 if no positive note exists, and 0 otherwise.

We then assess performance using this precision-level metric. We follow the same procedure as before, except that hyperparameter settings are now chosen on the basis of this precision metric on the validation set, instead of AUC. In the table below, we note the same trends are present with average precision as with AUC; namely, there is essentially no difference when all training data can be used, but a much larger difference when the model is restricted to only a fraction.

Table C.2: Average precision at 80% recall ± standard deviations (as calculated over five folds), comparing the use of all features to just those selected by OCP. The final row displays the difference in average precision between OCP-selected features and all features.

	Fraction of training data				
Available features	**1**	**1/2**	**1/4**	**1/8**	**1/16**
OCP subset	0.54 ± 0.09	0.54 ± 0.07	0.51 ± 0.08	0.48 ± 0.09	0.42 ± 0.10
All features	0.54 ± 0.12	0.50 ± 0.12	0.49 ± 0.05	0.37 ± 0.07	0.37 ± 0.07
OCP - All	0 ± 0.08	0.05 ± 0.10	0.02 ± 0.08	0.10 ± 0.07	0.05 ± 0.05

C.2.2 Nonlinear representations

Implementation Details

The masked language modeling was conducted using the BertForMaskedLM implementation from Wolf et al. [305] with a 15% masking rate and a learning rate of 5e-5. The checkpoint used downstream was selected as the one in which the model had the lowest validation loss on a held-out set of notes. The contrastive pre-training across all 3 objectives was conducted using the BertForNextSentencePrediction implementation from Wolf et al. [305] with a learning rate of 1e-5 and weight decay of 0.01. For each contrastive approach, the model checkpoint with the highest validation accuracy on a held-out set of 3200 contrastive pairs was chosen for use downstream. The L_2-regularized linear layer was implemented using scikit-learn, using class-balanced reweighting and the 'liblinear' optimizer. As before, the regularization (1e-2 to 1e5) was chosen for each seed, fold, model, and dataset size using the best performance on the validation set.

Computational Burden

Each pre-training method was trained on a single NVIDIA GeForce GTX 1080 Ti GPU with 12GB of memory. Convergence required about 6 hours for the masked language modeling pre-training and 3-4 hours for each of the contrastive pre-training methods. Since pre-training was conducted over 3 different seeds, it collectively required about 54 GPU hours. The forward pass to extract frozen embeddings from the different models across seeds and folds was minimal in time (less than 15 minutes). All downstream experiments and hyperparameter tuning involved solely a linear layer and were conducted on a CPU.

Granular Experimental Results

Below, we display more granular experimental results, splitting performance by seed. We display both the mean and standard deviation AUC across the 5 folds at each training data size. While there is some variation between seeds, we find that OCP consistently outperforms at smaller dataset sizes.

Table C.3: AUC average and standard deviation (as calculated over five folds) over each of the three seeds trained per method.

	Fraction of training data				
AUC ± std dev	**1**	**1/2**	**1/4**	**1/8**	**1/16**
OCP Seed A	0.87 ± .04	0.86 ± .04	0.84 ± .03	0.82 ± .03	0.83 ± .04
OCP Seed B	0.86 ± .03	0.85 ± .04	0.84 ± .04	0.81 ± .02	0.82 ± .03
OCP Seed C	0.88 ± .02	0.86 ± .04	0.85 ± .03	0.81 ± .03	0.79 ± .02
BERT	0.79 ± .06	0.74 ± .06	0.72 ± .04	0.63 ± .05	0.60 ± .06
Fine-Tuned LM Seed A	0.87 ± .05	0.85 ± .06	0.85 ± .02	0.77 ± .06	0.75 ± .06
Fine-Tuned LM Seed B	0.84 ± .03	0.82 ± .04	0.79 ± .02	0.70 ± .07	0.67 ± .09
Fine-Tuned LM Seed C	0.83 ± .02	0.80 ± .01	0.77 ± .01	0.73 ± .06	0.71 ± .05
Pt-Contrastive Seed A	0.82 ± .04	0.79 ± .03	0.77 ± .04	0.69 1 ± .07	0.66 ± .10
Pt-Contrastive Seed B	0.85 ± .02	0.83 ± .04	0.78 ± .04	0.72 ± .03	0.69 ± .1
Pt-Contrastive Seed C	0.84 ± .05	0.85 ± .05	0.83 ± .03	0.77 ± .04	0.72 ± .06
PCL Seed A	0.88 ± .02	0.88 ± .03	0.85 ± .02	0.83 ± .04	0.78 ± .03
PCL Seed B	0.86 ± .01	0.85 ± .04	0.80 ± .06	0.77 ± .02	0.76 ± .04
PCL Seed C	0.86 ± .03	0.84 ± .03	0.80 ± .04	0.75 ± .04	0.71 ± .08

Average Precision Results

Below we show results on the test set using the patient-level average precision metric introduced in Appendix B.1. As before, hyperparameter settings are chosen based on the model with best performance on the validation set, per this precision metric. Again, we find that OCP is relatively consistent in performance even with minimal training data, and outperforms other methods, particularly in the low data regime.

Table C.4: Performance in terms of average precision at 80% recall as calculated across 5 folds and 3 seeds for fine-tuned methods. The first row contains the mean average precision of OCP ± its standard deviation. The following rows contain the mean precision advantage of OCP over each comparison method, and the percentage of time OCP outperforms that method, computed over all seeds and folds.

	Fraction of training data				
Prec diff (OCP Win %)	**1**	**1/2**	**1/4**	**1/8**	**1/16**
OCP Prec	0.57 ± .08	0.58 ± .07	0.55 ± .06	0.50 ± .08	0.50 ± .09
OCP - BERT	0.16 (100%)	0.19 (100%)	0.19 (100%)	0.19 (100%)	0.22 (100%)
OCP - FT LM	0.09 (80%)	0.09 (80%)	0.09 (84%)	0.11 (87%)	0.13 (93%)
OCP - Pt-Contrastive	0.08 (73%)	0.09 (80%)	0.14 (100%)	0.15 (96%)	0.17 (98%)
OCP - PCL	0.02 (56%)	0.04 (69%)	0.07 (73%)	0.10 (84%)	0.13 (91%)

Qualitative Analysis

We now conduct a qualitative analysis to understand whether the OCP pre-trained model is operating as expected, namely whether it is attending to those features we would expect to be most crucial both in ordering and for downstream progression extraction.

The BertForNextSentencePrediction model used for contrastive pre-training is implemented such that sequence classification is based off of the representation of the `CLS` classifier token. Therefore, we examine those tokens which are most highly attended to by the `CLS` token in the last BERT layer, as a proxy signal for what the model is attending to for its final classification representation. Over all examples in the validation set, we find the average attention each token contributed in the final layer. There are then 21 tokens with an average attention above a threshold of 0.1; they include *increased/increase/increasing*, *change*, *unchanged*, *no*, *stable*, *negative*, *persistent*, and *resolved*. All such features elucidate disease stage, and therefore, it seems qualitatively plausible that OCP learns useful downstream representations in the nonlinear case. Other highly ranked features include the `SEP` token used to split the contrastive pairs, as well as *prior*, which is a possible leaky feature that indicates order information, but is not as useful for downstream analysis.

Appendix D

Additional Information for Chapter 7

D.1 Prompts and Sample GPT-3 Outputs

We present examples for each task alongside their corresponding prompts to illustrate different prompting strategies used for each task.

D.1.1 Clinical Sense Disambiguation

For clinical sense disambiguation we used GPT-3 edit mode which edits the input text rather than continuing it.

Example 1

GPT-3 edit input:

```
GENERAL: Patient is sedated on vent. HEENT: Normocephalic, atraumatic.
Pupils are sluggish but reactive and equal bilaterally. NECK: Supple. No
lymphadenopathy. No JVD. CARDIAC: Regular rate and rhythm. No murmurs.
LUNGS: CTA, intubated. ABDOMEN: Obese, nontender, positive bowel sounds.
EXTREMITIES: Positive pulses, positive edema. SKIN: Chronic changes
pretibial area bilaterally.
```

GPT-3 edit prompt:

```
Expand the abbreviation: CTA
```

GPT-3 edit output:

```
GENERAL: Patient is sedated on vent. HEENT: Normocephalic, atraumatic.
Pupils are sluggish but reactive and equal bilaterally. NECK: Supple. No
lymphadenopathy. No JVD. CARDIAC: Regular rate and rhythm. No murmurs.
LUNGS: Clear to auscultation, intubated. ABDOMEN: Obese, nontender,
positive bowel sounds. EXTREMITIES: Positive pulses, positive edema. SKIN:
Chronic changes pretibial area bilaterally.
```

Example 2

GPT-3 edit input:

```
2. Severe hypertension. 3. Severe mitral regurgitation. 4. Osteoporosis.
PROCEDURES: 1. Coronary angiography and hemodynamic evaluation with right
heart catheterization. Right heart catheterization shows right atrial
pressure of 1, right ventricular pressure of 47/11, PA pressure of 48/16,
with a pulmonary capillary wedge pressure of 29, with a large B-wave
confirmed with a wedge saturation of 95
```

GPT-3 edit prompt:

```
Expand the abbreviation: PA
```

GPT-3 edit output:

```
2. Severe hypertension. 3. Severe mitral regurgitation. 4. Osteoporosis.
PROCEDURES: 1. Coronary angiography and hemodynamic evaluation with right
heart catheterization. Right heart catheterization shows right atrial
pressure of 1, right ventricular pressure of 47/11, pulmonary artery
pressure of 48/16, with a pulmonary capillary wedge pressure of 29, with a
large B-wave confirmed with a wedge saturation of 95%.
```

D.1.2 Biomedical Evidence Extraction

For biomedical evidence extraction we used GPT-3 Davinci. The generated output is shown in blue.

Example 1

GPT-3 Davinci:

```
Assessment of acute bronchodilator effects from specific airway resistance
changes in stable COPD patients.
BACKGROUND In COPD patients, reversibility is currently evaluated from the
changes of forced expiratory volume at 1s (ΔFEV1) and forced vital capacity
(ΔFVC). By lowering peripheral airway smooth muscle tone, bronchodilators
should decrease dynamic hyperinflation, gas trapping, and possibly dyspnea
at rest. Hence, we hypothesize that specific airway resistance changes
(ΔsRAW) should better characterize the acute response to bronchodilators.
METHODS On two days, 60 COPD patients underwent dyspnea evaluation (VAS
score) and pulmonary function testing at baseline and one hour after
placebo or 300μg indacaterol administration.
RESULTS Spirographic and ΔsRAW-based criteria identified as responders 24
and 45 patients, respectively. ΔsRAW correlated with changes of
intrathoracic gas volume (ΔITGV) (r=0.61; p<0.001), residual volume (ΔRV)
(r=0.60; p<0.001), ΔFVC (r=0.44; p=0.001), and ΔVAS (r=0.73; p<0.001),
while ΔFEV1 correlated only with ΔFVC (r=0.34; p=0.008). Significant
differences in terms of ΔITGV (p=0.002), ΔRV (p=0.023), and ΔVAS (p<0.001)
occurred only if patients were stratified according to ΔsRAW. [...]

Create a bulleted list of the arms in this trial.

- Placebo
- Indacaterol (300 μg)
```

Example 2

GPT-3 Davinci:

Serum biochemical characteristics of Beluga, Huso huso (L.), in response to blood sampling after clove powder solution exposure.
In order to investigate the effect of anesthesia on serum parameters, Beluga, Huso huso (L.) were blood-sampled immediately without anesthesia (control) or subjected to following anesthesia procedure: 40, 120, and 240 s exposure to 3,000, 700, and 500 mg l^{-1} clove solution, respectively. Blood samples were collected after these periods, when fish were immobile and reached stage 4 anesthesia. Results showed that cortisol and glucose levels were significantly high in 700 and 500 but not 3,000 mg l^{-1} group compared to control. Serum lactate levels were significantly high in 500 mg l^{-1} group compared to control group. Lactate levels were not significantly differed between control, 3,000, and 700 mg l^{-1} groups. There were no significant differences in serum levels of cholesterol, total protein, lactate dehydrogenase, aspartate aminotransferase, alanine aminotransferase, Na^{+}, Cl^{-}, K^{+}, and Ca^{2+}. Results suggest that rapid anesthesia with higher dose is better than slow anesthesia with lower dose for blood sampling in Beluga.

Create a bulleted list of the arms in this trial.

- Control
- 3,000 mg l^{-1} clove solution
- 700 mg l^{-1} clove solution
- 500 mg l^{-1} clove solution

D.1.3 Coreference Resolution

Example 1: Zero-shot unguided

GPT-3 Davinci:

```
[...] Her current regimen for her MS is Rebif Monday, Wednesday, and Friday
and 1 gram of methylprednisolone p.o. every month. This had been working
previously; however, she feels that her symptoms return before her next
dose of methylprednisolone is due.

What does "This" in the sentence "This had been working previously; however,
she feels that her symptoms return before her next dose of
methylprednisolone is due." refer to?

The sentence is unclear, but it seems to be referring to the Rebif regimen.
```

Example 1: One-shot guided (incorrect)

```
[...] ASSESSMENT & PLAN: The patient has no physical evidence of
progression of disease. It is somewhat worrisome that her CEA is up, but
will be keep watching that [...]

What does "that" in the sentence "It is somewhat worrisome that her CEA is
up, but will be keep watching that to see if it is just within a normal
swing for her." refer to?

"that" refers to "progression of disease"

[...] Her current regimen for her MS is Rebif Monday, Wednesday, and Friday
and 1 gram of methylprednisolone p.o. every month. This had been working
previously; however, she feels that her symptoms return before her next
dose of methylprednisolone is due.

What does "This" in the sentence "This had been working previously; however,
she feels that her symptoms return before her next dose of
methylprednisolone is due." refer to?

"This" refers to "her current regimen for her MS"
```

Example 1: One-shot guided (correct)

```
[...] ASSESSMENT & PLAN: The patient has no physical evidence of progression
of disease. It is somewhat worrisome that her CEA is up, but will be keep
watching that to see if it is just within a normal swing for her. [...]

What does "that" in the sentence "It is somewhat worrisome that her CEA is
up, but will be keep watching that to see if it is just within a normal
swing for her." refer to?

"that" refers to "her CEA"

[...] Her current regimen for her MS is Rebif Monday, Wednesday, and Friday
and 1 gram of methylprednisolone p.o. every month. This had been working
previously; however, she feels that her symptoms return before her next
dose of methylprednisolone is due.

What does "This" in the sentence "This had been working previously; however,
she feels that her symptoms return before her next dose of
methylprednisolone is due." refer to?

"This" refers to "her current regimen for her MS"
```

D.1.4 Medication Status Extraction

Example 1: Zero-shot guided

```
[...] home dose of Kadian as this is her long-acting medication and DC the
continuous Dilaudid given IV. 5. Urinary tract infection with Klebsiella
and E. coli, both sensitive to Levaquin. Since this was diagnosed Foley has
been DC’d. For now would continue Levaquin and recheck urinalysis.

Create a bulleted list of which medications are mentioned and whether they
are active, discontinued, or neither.

-”Kadian” (active)
-”Dilaudid” (discontinued)
-”Levaquin” (active)
```

Example 1: One-shot guided (incorrect)

```
[...] start her on Ativan 1 mg p.o. q. 8 hours and use Ativan 1 mg IV q. 4
hours p.r.n. for agitation. I will also start her on Inderal LA 60 mg p.o.
q.d. for essential tremors. She does not want to take Celexa, and I will
put her back on Lexapro 2 mg p.o. q.d. I will discontinue Esmolol.

Create a bulleted list of which medications [...]

-”Ativan” (discontinued)
-”Inderal LA” (active)
-”Celexa” (neither)
-”Lexapro” (active)
-”Esmolol” (active)

[...] home dose of Kadian as this is her long-acting medication and DC the
continuous Dilaudid given IV. 5. Urinary tract infection with Klebsiella
and E. coli, both sensitive to Levaquin. Since this was diagnosed Foley has
been DC’d. For now would continue Levaquin and recheck urinalysis.

Create a bulleted list of which medications [...]

-”Kadian” (active)
-”Dilaudid” (discontinued)
-”Levaquin” (active)
```

Example 1: One-shot guided (correct)

```
[...] start her on Ativan 1 mg p.o. q. 8 hours and use Ativan 1 mg IV q. 4
hours p.r.n. for agitation. I will also start her on Inderal LA 60 mg p.o.
q.d. for essential tremors. She does not want to take Celexa, and I will
put her back on Lexapro 2 mg p.o. q.d.

Create a bulleted list of which medications [...]

-"Ativan" (active)
-"Inderal LA" (active)
-"Lexapro" (active)
-"Celexa" (neither)

[...] home dose of Kadian as this is her long-acting medication and DC the
continuous Dilaudid given IV. 5. Urinary tract infection with Klebsiella
and E. coli, both sensitive to Levaquin. Since this was diagnosed Foley has
been DC'd. For now would continue Levaquin and recheck urinalysis.

Create a bulleted list of which medications [...]

-"Kadian" (discontinued)
-"Dilaudid" (discontinued)
-"Levaquin" (discontinued)
```

D.1.5 Medication Attribute Extraction

Example 1: Token-level

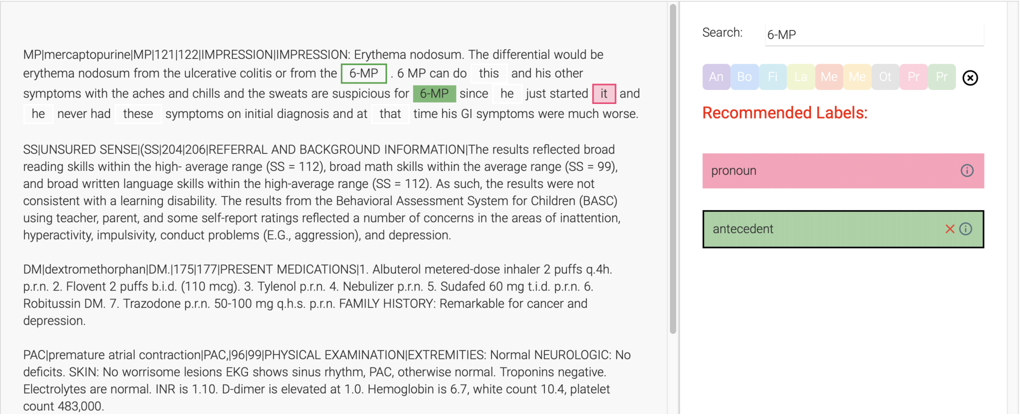
```
<One-shot-example>

Label the tokens: medication, dosage, route, frequency, duration, reason,
or none. Ignore allergies.

<One-shot-labels>

Input: 8. Albuterol 2 puffs every 4-6 hours as needed. HOSPITAL COURSE:
This is an 80-year-old female who was hospitalized about 2 months ago for
chronic obstructive pulmonary disease exacerbation. At that time she was
put on prednisone and antibiotics and seemed to get better. However, she
was put on Augmentin ES and continued to have difficulty tasting food and
felt that food tasted very salty. She had no appetite and she has continued
to lose weight over the last 2 months.

Label the tokens: medication, dosage, route, frequency, duration, reason,
or none. Ignore allergies.

-"8": none
-".": none
-"Albuterol": medication
-"2": dosage
-"puffs": dosage
-"every": frequency
-"4-6": frequency
-"hours": frequency
-"as": none
-"needed": none
-".": none
-"HOSPITAL": none
-"COURSE": none
-"This": none
-"is": none
-"an": none
-"80-year-old": none
-"female": none
-"who": none
-"was": none
-"hospitalized": none
-"about": duration
-"2": duration
-"months": duration
-"ago": duration
-"for": reason
-"chronic": reason
-"obstructive": reason
[...]
```

Example 1: Phrase-level

```
<One-shot-example>

Label the tokens: medication, dosage, route, frequency, duration, reason,
or none. Chunk the same entity together. Ignore allergies.

<One-shot-labels>

Input: 8. Albuterol 2 puffs every 4-6 hours as needed. HOSPITAL COURSE:
This is an 80-year-old female who was hospitalized about 2 months ago for
chronic obstructive pulmonary disease exacerbation. At that time she was
put on prednisone and antibiotics and seemed to get better. However, she
was put on Augmentin ES and continued to have difficulty tasting food and
felt that food tasted very salty. She had no appetite and she has continued
to lose weight over the last 2 months.

Label the tokens: medication, dosage, route, frequency, duration, reason,
or none. Chunk the same entity together. Ignore allergies.

-”8”: none
-”.”: none
-”Albuterol”: medication
-”2 puffs”: dosage
-”every 4-6 hours”: frequency
-”as needed”: duration
-”.”: none
-”HOSPITAL COURSE”: none
-”This”: none
-”is”: none
-”an”: none
-”80-year-old”: none
-”female”: none
-”who”: none
-”was”: none
-”hospitalized”: none
-”about”: none
-”2 months”: duration
-”ago”: none
-”for”: none
-”chronic”: none
-”obstructive”: none
-”pulmonary”: none
-”disease”: reason
-”exacerbation”: none
-”.”: none
-”At”: none
-”that”: none
[...]
```

Example 1: Relation Extraction

```
 <One-shot-example>

Label medications, ignoring allergies. Include dosage, route, frequency,
duration, reason, if available.

<One-shot-labels>

Input: 8. Albuterol 2 puffs every 4-6 hours as needed. HOSPITAL COURSE:
This is an 80-year-old female who was hospitalized about 2 months ago for
chronic obstructive pulmonary disease exacerbation. At that time she was
put on prednisone and antibiotics and seemed to get better. However, she
was put on Augmentin ES and continued to have difficulty tasting food and
felt that food tasted very salty. She had no appetite and she has continued
to lose weight over the last 2 months.

Label medications, ignoring allergies. Include dosage, route, frequency,
duration, reason, if available.

-"medication: "Albuterol”, dosage: "2 puffs”, frequency: "every 4-6 hours”,
duration: "as needed”
-medication: "prednisone”, duration: "2 months”
-medication: "antibiotics”, duration: "2 months”
-medication: "Augmentin ES”, duration: "2 months”
```



MP|mercaptopurine|MP|121|122|IMPRESSION|IMPRESSION: Erythema nodosum. The differential would be erythema nodosum from the ulcerative colitis or from the 6-MP . 6 MP can do this and his other symptoms with the aches and chills and the sweats are suspicious for 6-MP since he just started it and he never had these symptoms on initial diagnosis and at that time his GI symptoms were much worse.

SS|UNSURED SENSE|(SS|204|206|REFERRAL AND BACKGROUND INFORMATION|The results reflected broad reading skills within the high- average range (SS = 112), broad math skills within the average range (SS = 99), and broad written language skills within the high-average range (SS = 112). As such, the results were not consistent with a learning disability. The results from the Behavioral Assessment System for Children (BASC) using teacher, parent, and some self-report ratings reflected a number of concerns in the areas of inattention, hyperactivity, impulsivity, conduct problems (E.G., aggression), and depression.

DM|dextromethorphan|DM.|175|177|PRESENT MEDICATIONS|1. Albuterol metered-dose inhaler 2 puffs q.4h. p.r.n. 2. Flovent 2 puffs b.i.d. (110 mcg). 3. Tylenol p.r.n. 4. Nebulizer p.r.n. 5. Sudafed 60 mg t.i.d. p.r.n. 6. Robitussin DM. 7. Trazodone p.r.n. 50-100 mg q.h.s. p.r.n. FAMILY HISTORY: Remarkable for cancer and depression.

PAC|premature atrial contraction|PAC,|96|99|PHYSICAL EXAMINATION|EXTREMITIES: Normal NEUROLOGIC: No deficits. SKIN: No worrisome lesions EKG shows sinus rhythm, PAC, otherwise normal. Troponins negative. Electrolytes are normal. INR is 1.10. D-dimer is elevated at 1.0. Hemoglobin is 6.7, white count 10.4, platelet count 483,000.

Search: 6-MP

Figure D-1: Platform used for annotation of the three new labeled datasets, shown for the coreference resolution annotation task.

D.2 Annotation Process

A screenshot of the annotation process can be seen in Figure 2.

D.2.1 Biomedical Evidence Extraction

For arm identification 20 out of 187 abstracts from the test set of Nye et al. [196] were randomly selected. Both annotators separately identified the study arms in each abstract. They included all characteristics of a arm that were necessary for differentiation [71]. For example, we would not require the route of administration for a drug (e.g., “oral" in *oral X*) unless another arm contained the the same drug in a different formal (e.g., *X nasal spray*). There was full consensus between annotators for the identified numbers of arms. A single abstract was replaced due to its ambiguity.

D.2.2 Coreference Resolution

Annotators labeled 105 snippets from the CASI dataset with pronouns and their corresponding noun phrase antecedent [176]. The antecedent was annotated as the entire noun phrase (barring any dependent clauses); in cases where two antecedents were available, both were labeled. For the purposes of evaluation, we chose the antecedent with the highest overlap to each model's output. To ensure nontrivial examples, the annotators excluded all examples of personal pronouns (e.g., "he", "she") if another person (and possible antecedent) had not yet been mentioned in the snippet.

D.2.3 Medication Status Extraction

We wanted to create a dataset of challenging examples containing a changeover in treatment. From a sample, only ∼5% of CASI snippets contained such examples. To increase the density of these examples, speeding up annotation, clinical notes were filtered with the following search terms: *discont*, *adverse*, *side effect*, *switch*, and *dosage*, leading to 1445 snippets. We excluded snippets that were purely medication lists, requiring at least some narrative part to be present. For 105 randomly selected snippets, the annotators first extracted all medications. Guidelines excluded medication categories (e.g., "ACE-inhibitor") if they referred to more specific drug names mentioned elsewhere (even if partially cut off in the snippet). For instance, only the antibiotic Levaquin was labeled in: "It is probably reasonable to treat with antibiotics [...]. I would agree with Levaquin alone [...]". Guidelines also excluded electrolytes and intravenous fluids as well as route and dosage information. In a second step, medication were assigned to one of three categories: *active*, *discontinued*, and *neither*. Discontinued medications also contain medications that are temporarily on hold. The category *neither* was assigned to all remaining medications (e.g., allergies, potential medications).

D.2.4 Medication Attribute Extraction

For medication attribute extraction, we also labeled 105 examples from CASI [176]. Annotation guideline were adopted from the 2009 i2b2 medication extraction challenge [276] with slight modifications. We allowed medication attributes to have multiple spans. Also, we grouped together different names of the same drug (e.g., "Tylenol" and "Tylenol PM") for the purpose of relation extraction. After annotation of the data, we create three versions of the dataset: token-level, phrase-level, and relation-level.

For the first, we split all word in the example and assigned them their respective label or *none* if they were not part of a label (see token-level example in D.1.5. For phrase-level, we kept consecutive words with the same label grouped together as phrases (see phrase-level example in D.1.5. The relation level just contained the extracted medication and their attributes (see relation extraction example in D.1.5. We note that medication lists were downsampled in the creation of the dataset, since the 2009 i2b2 challenge had found performance on narrative text was far lower than on medication lists.

D.2.5 Clinical Sense Disambiguation

How do we know CASI is not in the LLM training set? Since the CASI dataset is publicly accessible from the Internet and on Github, one potential pitfall is that the dataset may have been in the language models' training data. While this is also true of other common NLP benchmarks, we attempted to confirm results were not merely an artifact. To do so, we annotated 50 distinct acronyms that occurred in sentences in the CASI dataset that were *not* included in the original annotations. While this set of acronyms is easier (e.g., they many only have a single clinical expansion), this allows us to check that GPT-3 is not simply pattern matching to potential past training data. In the set of 50, we find *GPT-3 edit* correctly expanded 47 (94%). In 2 of these cases, the acronym was in fact a typo (SMIV instead of SIMV, AVG instead of ABG), and the correct expansion was given regardless. Of the 3 that were incorrect, one was in fact incorrect, one was of unspecified meaning to the annotator, and one had 2/3 of the words correct in the expansion. **Resolver Details**

Weak Supervision For weak supervision, we only consider the 97% of the dataset where the overlap with an answer choice was at least 5 characters as candidates for pseudolabels. Following prior work [143, 144], we additionally used a technique called the *cut statistic* to select a high-quality subset of the weakly labeled data to reduce the noise in the training process. We selected a subset of size 75% to decrease noise while still choosing a large enough set to ensure all acronyms were seen during training. We fine-tuned a PubMedBERT [96] model, a BERT variant that was pretrained on biomedical abstracts and full-text articles from PubMed, using learning rate 1e-5, weight decay 0.01, the AdamW optimizer [163], and batch size 4, using the `BERTForMultipleChoice` functionality in HuggingFace Transformers [304].

D.2.6 Biomedical Evidence Extraction

Baseline Training Details We trained for 10,000 steps using the AdamW optimizer, learning rate 2e-5, batch size 32, and weight decay 1e-6, inheriting these hyperparameters from Zhang et al. [320]. These were the best-performing hyperparameters across the set reported in Zhang et al. [320, Table 10, "BERT-CRF"].

Resolver Details To evaluate on the original token-level labels we tokenize the GPT-3 output and remove bullet points, numbers, stop words, and the words "treatment", "control", and "group" which GPT-3 often appended for clarification (e.g., "- Placebo (Control group)"). Then, any token in the input that is found in the remaining GPT-3 output is labeled with a 1, and others with a 0. Since our procedure may have interrupted valid spans, we fill in any 0's between 1's as well as acronyms within parentheses. These steps transform the LLM output strings l_i to a binary labeling of the full input.

Example of Token-level Error Modes As an example describing token-level error modes of GPT-3, consider the output, the resolved output, and the gold label for a study with two arms below.

GPT-3 output

- *Inhaled fluticasone*
- *Placebo*

Resolved GPT-3 output:

Inhaled fluticasone reduces [...] double-blind, placebo-controlled study [...] inhaled fluticasone [...] or placebo. Large-scale [...] of inhaled steroid therapy on [...]

Gold-label (token-level):

Inhaled fluticasone reduces [...] double-blind, placebo-controlled study [...] inhaled fluticasone [...] or placebo. Large-scale [...] of inhaled steroid therapy on [...]

GPT-3 correctly identifies both study arms. However, the resolved output, which simply labels the token sequence of the identified arms in the original input, disagrees with the gold labels for several tokens. For example, the output includes the route, "inhaled", which isn't kept in the annotation schema, dinging precision. Further, the output excludes “placebo-controlled” (given “placebo” is included), dinging recall. Therefore, despite qualitatively capturing the arms of this trial, there was a middling F1-score of 0.70 for this example. This serves to underline why token-level metrics can be misleading as to true performance towards the underlying goal.

Oracle Details We assumed oracle splitting and oracle coreference resolution in order to distill the token-level labels to a list for the PubMedBERT baselines. As an example of oracle splitting, PubMedBERT assigned a 1 to the span *“40, 120, and 240 s exposure to 3,000, 700, and 500mg* l^1 *clove solution;”* this span in fact contains three different arms, and we assume it can be perfectly split, since the required information is theoretically present in the identified span. As an example of oracle coreference resolution, consider this example with two arms: *capecitabine and oxaliplatin plus radiotherapy (Cap-Oxa-CRT)* and *concurrent capecitabine and radiotherapy (Cap-CRT)*. The spans recognized by PubMedBERT include “adjuvant concurrent chemotherapy”, “capecitabine-based concurrent chemotherapy”, “postoperative CRT of capecitabine with or without oxaliplatin”, “concurrent capecitabine and radiotherapy (Cap-CRT)” and “capecitabine and oxaliplatin plus radiotherapy (Cap-Oxa-CRT).” To be generous to the baseline, we assumed those 5 spans *could* possibly be reduced to the two arms with oracle coreference resolution. No oracle splitting or coreference resolution was conducted for Resolved GPT-3.

Analysis of Error Modes for Arm Identification Resolved GPT-3 successfully identified the correct number and content of the arms in 17 of the 20 examples. The three examples it missed were also missed by PubMedBERT. In one case with two arms, both methods included a procedure as a separate third arm; in reality, the procedure occurred for both arms and was not the intervention itself. In the second case, the prompt output did not elaborate on the treatment group sufficiently, and in the final case, it fully misparsed. Assuming the oracle splitting and coreference, PubMedBERT would still have issues with 10 further examples: two again included a common procedure as a third arm, four were missing control arms, one was missing a treatment arm, two arms required further domain knowledge to consolidate (e.g., that Ramipril is an ACE inhibitory therapy), and another required properly consolidating a therapy with no overlapping tokens.

D.2.7 Coreference Resolution

Baseline Details We benchmark using a transformer-based model trained jointly on three large coreference datasets [274] that can be found on the HuggingFace model hub (`shtoshni/longformer_coreference_joint`).

Resolvers The resolver for the 0-shot unguided prompt was 50 LOC, or 973 tokens in the Codex tokenizer. In contrast, the 1-shot guided prompt required only stripping a final quotation mark, period, or space, which required 20 tokens per the Codex tokenizer.

D.2.8 Medication + Status Extraction

Resolver details For an unguided prompt, to map the GPT-3 output string to a list of medication strings, the first step is to break the output string up into substrings by parsing the "bulleted list" output by GPT-3, which we do with regular expressions. The output strings for this prompt followed several different formats, making this step slightly more involved than in previous cases. The two basic formats were a newline-separated list and a comma-separated list of medication names. The modifiers were also expressed in different ways: some outputs were *{Medication}: {Status}*, while others were *{Medication} ({Status})*. A few examples instead grouped the medications by status, so the output was *Active: {medication1}, {medication2}, Discontinued: {medication3}*. Examples of these outputs can be found in Appendix D.1.4. Despite this variation, we output a list by simply replacing newlines with commas to reduce to the comma-separated case, and then applying two regular expressions to extract the medication names and modifiers from the list.

The previous steps turn the LLM output strings into lists of strings. The next step in the resolver is to *denoise* the individual strings in each list by first stripping dosage and route information (e.g., “10 mg” or “patch”) and then performing input-consistency checking by removing tokens that do not appear in the input. Finally, strings that, after the prior denoising steps, only consist of stop words or primarily consist of punctuation and whitespace, are removed from the prediction lists. This required 32 lines of code, and 946 tokens in a byte-pair encoding. In contrast, with a 1-shot prompt, output could be simply split on the bullets, and the status extracted from parentheses, requiring 8 lines of code and 165 tokens in a byte-pair encoding.

Medication Extraction Baseline For normalization, all entities were linked to the UMLS via the default string overlap functionality of ScispaCy [31]. We filtered the resulting UMLS concepts by their semantic types and only kept concepts of the types *Antibiotic*, *Clinical Drug*, *Pharmacologic Substance*, and *Vitamin.* Finally, the baseline predictions are run through the same denoising steps as the GPT-3 predictions to ensure a fair comparison.

Status Classification: T-Few We use T-Few [159] for medication status classification using 20 additional annotated examples as the few-shot training set. We used a single prompt:

```
In the clinical note below, what is the status of the medication Albuterol?

Albuterol 2 puffs every 4-6 hours as needed. HOSPITAL COURSE: This is an
80-year-old female who was hospitalized about 2 months ago for chronic
obstructive pulmonary disease exacerbation. At that time she was put on
prednisone and antibiotics and seemed to get better. However, she was put
on Augmentin ES and continued to have difficulty tasting food and felt that
food tasted very salty. She had no appetite and she has continued to lose
weight over the last 2 months.
```

For the answer choices, we used `Discontinued`, `Active`, and `Neither`. We did not use $\mathrm{IA}^{(3)}$ pretraining, but otherwise directly followed the T-Few recipe (i.e., we used the default values for all hyperparameters including batch size, learning rate, number of steps, length normalization, etc.). We used the T0-11B model.

D.2.9 Medication + Relation Extraction

Resolver The resolver for the first two tasks iterates over the lines in the GPT-3 output and grabs both the text span and the label; the text span is mapped to tokenized space, and all labels not in the label space (e.g., "Instructions") are mapped to *None*. For phrase-level labeling, a single additional step is conducted to map the labels to BIO format. For the relation extraction task, the resolver additionally assumes all entities mentioned in a line correspond to the medication on that line.

Sequence Tagging baseline We model extraction and labeling of medication + modifier (dosage, frequency, route, duration, reason) as a sequence tagging task. We use the `B/I/O` encoding for the label space, adding tags to the `B` and `I` labels indicating the type of entity. For training data, we split the 10 notes from the 2009 i2b2 challenge into shorter contexts using an off-the-shelf sentence segmenter, and merged split contexts of less than 30 tokens into the previous context. This results in 176 training contexts for the PubMedBERT + CRF model. As with Biomedical Evidence Extraction, we search for hyperparameters over the search space reported in Zhang et al. [320, Table 10, "BERT-CRF"]. The final model is chosen based on validation F1 score on a randomly selected validation set of 10% of the training data (i.e., 18 contexts).

Relation Extraction Baseline We use the model from Shi and Lin [247] for relation extraction on top of PubMedBERT. For training data, we again use the 2009 i2b2 challenge set, but since the goal is to associate modifiers with individual medications, we split up the 10 long notes into rolling chunks around each medication mention. For each ground-truth medication entity, we create a context including the 30 tokens before and after that entity. We extended these windows to be on an `O` label so that entities are not split across contexts. We use a binary label space, since each modifier type (dosage, route, etc.) determines the relation type: the relevant task is to classify whether each pair of (medication, modifier) entities in a span is associated. We create one positive sample for each truly related (medication, modifier) pair. For each context, we add a negative sample for each (medication, modifier) pair that is not related. This results in 1416 examples (many of which have largely overlapping context, but a different pair of entities) for training the relation extraction model.

Task	**Cost per Token**	**Tokens per Example**	**# of Examples**	**# of Settings**	**Approx. Cost**
Clinical sense disambiguation	$0 (free in *edit* beta mode)	100	105	1	$0
Biomedical evidence extraction	$0.00006	500	187	1	$6
Coreference resolution	$0.00006	300	105	11	$21
Medication status extraction	$0.00006	300	105	16	$30
Medication attribute extraction	$0.00006	600	105	3	$12

Table D.1: Estimate of cost of running the experiments included in this chapter

D.3 Experimental Cost

At time of experimentation the cost of experiments included in this work were under $100. A breakdown of the upper bound of API costs can be found in Table D.1 and is based on OpenAI API pricing in spring 2022. All estimates of tokens/example are rough upper bounds; some experimental settings were cheaper.

Research Article	2992
Patient Health Resource	1975
Commercial Health	1317
Medical Encyclopedia/Dictionary	499
Clinician Forum	404
Personal Blog	277
News Article	269
Patient Forum	179
Other Health	65
Non-Health	631

Table D.2: Number of search results in each category of the 8608 scraped documents, as classified by `text-davinci-003`.

D.4 Website Category Classification

D.4.1 Accuracy

The distribution of website category counts out of 8608 can be found in Table D.2. While not an exhaustive evaluation, we find that the website classifications from `text-davinci-003` generally match expectation. Details are below:

- The term "forum" appeared in 30% of the URLs for the `Patient Forum` class and 22% for `Clinician Forum`; it otherwise peaked at 1.5% in the `Other Health` category. The clinician forum URL often dealt with specific professional organizations (e.g., American College of Physicians), whereas patient forum URLs often mentioned specific diseases.
- The term "blog" appeared in 32% of URLs for the `Personal Blog` class; it otherwise peaked at 5% otherwise, in the `Non-Health` class). Additionally, terms denoting blog servers (e.g., "blogspot" "wordpress") were common as well as words pertaining to family (e.g., "mom").
- The term "journal" appeared in 10% of the URLs for the `Research Article`; it otherwise peaked at 1% in the `Personal Blog` category. The `Research Article` further captured 109/110 URLs containing "PubMed."
- The term "news" appeared in 53% of the URLs for the `News Article`; it otherwise peaked at 3% in the `Non-Health` category. We do note some fuzziness here in boundaries, as e.g., some patient health resources contain news pages as well.

- The term "WebMD" appeared in just 2% of URLS for `Patient Health Resources`, though 91% of the "WebMD" mentions occurred in the class, confirming high positive predictive value.
- The term "dict" appeared in 16% of the URLS for `Medical Dictionary/Encyclopedia`; it otherwise peaked at 1.6% for `Patient Health Resource`.
- The term "product" appeared in 16% of the URLS for the `Commercial Health` category; it otherwise peaked at 2% for `Medical Dictionary/Encyclopedia`.
- The `Other health` category was dominated in plurality by "Patents", comprising 30% of URLs. The term did not appear more than 0.2% in any other category.
- The `Non-health` had no single dominating thread, as one could predict.

D.4.2 Examples

Example scraped data from the eight major classes can be found in Table D.3, alongside the root source website. For patient privacy, exact links are not included but are available upon request.

	Example Source	Example Quote
Research Article	BioMed Central	“Post-surgery chemotherapy consisted of carboplatin (CARBO)”
Patient Health Resource	CancerGRACE	“the combination of carboplatin/ taxol (paclitaxel) with ... longer survival than carbo/taxol alone”
Commercial Health	Pharmacy Website	“Exporter of Carboplatin Injection... known as CARBO”
Medical Encyclopedia/ Dictionary	Wikipedia	“Subcutaneous administration may be abbreviated as SC, SQ, sub-cu, sub-Q, SubQ, or subcut.”* (No relevant carbo results in first 100)
Clinician Forum	OncLive	“ the weekly CARBO [carboplatin]/ paclitaxel regimen is”
Personal Blog	Wordpress	“the carboplatin part.... my first cycle of Carbo-Taxol”
News Article	ESMO Oncology News	“has approved pembrolizumab ...in combination with pemetrexed and carboplatin (pem/carbo)”
Patient Forum	Cancer Survivors Network	“anyone have experience with Carboplatin?” “carbo is easier than”

Table D.3: Examples for the eight major categories (excluding other/non-health) of websites containing medical jargon. Examples shown for carbo/carboplatin, except for the `Medical Encyclopedia/Dictionary` option.

Appendix E

Additional Information for Chapter 8

E.1 Data Extraction and Featurization

E.1.1 Examples of Clinical Notes

Here, we show examples of a triage note, chief complaint, patient vitals, and a clinician note. To preserve patient privacy, these examples are synthetic but mimic the formatting and style of real data.

Triage Note

```
pt with ruq abd pain and nonproductive cough
```

Chief Complaint

```
ruq abd pain
```

Vitals

```
Blood Pressure: 140/90 mmHg

Heart Rate: 109 BPM

Pain: 8 (out of 10)

Sex: F

Age: 66

Respiratory Rate: 92%
```

Temperature: 99 (deg. Fahrenheit)

Pulse Oxygen (Oxygen Saturation): 96

Clinical Note

HPI: 66 y/o F p/w ruq abd pain and nonproductive cough.
No fever, nausea, or chills.
History of chronic abdominal pain over last 4-5 years,
as well as htn and dmii.

PMH: htn, dmii, chronic abdominal pain, hysterectomy in 2004

MEDICATIONS: metoprolol tartrate, metformin

FAMILY HISTORY: Diabetes in mother,
father (deceased) hypertensive

SOCIAL HISTORY: no smoking, drinks socially

REVIEW OF SYSTEMS:
Constitutional - no fever, chills, nausea
Head / Eyes - no diplopia
ENT - no earache
Resp - nonproductive cough, mild
Cards - no chest pain
Abd - ruq abd pain
Flank - no dysuria
Skin - no rash
Ext - no back pain
Neuro - no headache
Psych - no depression

```
PHYSICAL EXAM: Ruq abd pain, tender to touch,
with some bloating.

MDM:
66 y/o F p/w ruq abd pain and mild cough. She reports
she had a cold last week, so cough
is likely symptom of that.

Epigastric pain with mild bloating and minor
heartburn. Gave an antacid to relieve pain.

Glucose levels are elevated compared to baseline
(140 6 hours ago, 120 averaged over last six months).
Says she will work on controlling diet more.

DIAGNOSIS: epigastric pain/heartburn
```

E.1.2 The NegEx Algorithm

We use a version of the NegEx algorithm [45] in order to perform a rule-based negation detection on clinical text. The algorithm greedily iterates through words in a piece of text and assigns them to a negated context if they are preceded by predefined keyword triggers. Pseudocode for the algorithm is shown in Figure E-1.

E.1.3 Trie-Based Extraction of UMLS Concepts

In order to confirm that our UMLS-mapped trie-based extraction of clinical concepts was reasonably accurate and performant, we also consider a few alternate ways of perform clinical NER on ED note text. We restrict our search to techniques that

```
fullstops = ['.', '-', ';']
midstops = ['+', 'but', 'and', 'pt', '.', ';', 'except',
            'reports', 'alert', 'complains', 'has',
            'states', 'secondary', 'per', 'did', 'other',
            'p/w', 'presents', 'presenting', 'presented', ':']
negwords = ['no', 'not', 'denies', 'without', 'non', 'lack']

def negation_detection(words):
    flag = 0
    res = []
    for i, w in enumerate(words):
        neg_start_condition = (flag == 1)
        neg_stop_condition =  (w in fullstops + midstops + negwords) or
                 (i > 0 and words[i-1][-1] in (fullstops + ['\n']))
        neg_end_of_list = (i==(len(words)-1) )
        if neg_start_condition and neg_stop_condition:
            flag = 0
            res += [(start_index, i-1)]
        elif neg_start_condition and neg_end_of_list:
            flag = 0
            res += [(start_index, i)]
        if w in negwords:
            flag = 1
            start_index = i
    return res
```

Figure E-1: Pseudocode of the rule-based negation detection algorithm.

normalize to UMLS, as this is a key benefit of our system that makes it extendable.

First, we attempted to extract concepts directly from the raw text, without normalizing to an ontology. We did this by extracting common unigrams and bigrams and removing common stopwords (`and`, `to`). We manually went through the 1,000 most common terms to confirm they were reflected in our UMLS-mapped ontology of conditions, and added a handful of terms that were missing: `hld` as a synonym for hyperlipidemia, `hep c` as a synonym for hepatitic C, `pna` for pneumonia, etc. We note that ontologies are always a work in progress and that our current system provides doctors with the ability to submit ontology modifications that can then be reviewed.

We compare our trie-based extraction against three baselines:

- `cTakes`, or the Mayo clinical Text Analysis and Knowledge Extraction System, which combines rule-based and simple machine learning techniques to extract and normalize concepts to UMLS [238]. cTakes is an older system that often misses clinical abbreviations [225]. We limit the cTakes vocabulary to UMLS concepts in our ontology to provide a fair comparison.

- `scispaCy`, which is a Python biomedical text processing library built on top of `spaCy` [190]. It contains neural entity extraction trained on biomedical corpora using a bidirectional-LSTM with a conditional random field (CRF) layer as proposed in [142]. `scispaCy` identifies clinical and biomedical terms on the text first with its entity recognition model, and then retroactively maps this to UMLS using a string match over synonyms.

- BERT-based clinical entity extraction models such as [10], which combine a transformer architecture with CRFs and other layers that are good at entity identification. These models are considered state-of-the-art in neural entity extraction, but are fairly slow and cannot easily run on our servers, which we discuss below. While we cannot easily compare to [10] due to the lack of labelled data to train the deep model, we measure latency of running BERT on a sequence of clinical notes as a proof-of-concept. We use DistilBERT as our base BERT model because of its compactness [232], and train on a custom vocabulary which

System	Latency (seconds)	Comments
Trie-based	0.8	Ours, poor disambiguation for the few overloaded concepts
cTakes	37	Provides virtually the same extraction as the trie-based procedure, but with certainty/polarity scores
scispaCy	19.5	Bulk of the time spent on mapping extracted terms to UMLS. Some acronyms were not disambiguated, e.g., `dm` was extracted as both `diabetes mellitus` and `double miutes`
DistilBERT	489	No extraction, just passing windowed snippets of the text through a compact transformer

Table E.1: Comparing NER approaches on OMR notes both by latency and by qualitative ability to extract concepts well. Latency is measured by time to process 100 randomly chosen OMR notes.

is smaller than that of the original BERT model [61].

While it is difficult to quantitatively compare these methods because we lack gold-standard entity labels for our dataset, we find that the trie-based method is significantly faster than our three other comparisons with little to no loss in recognition quality.

Note that all of the learned models also preclude us from making easy changes to our ontology– it is difficult to retrain these models without sufficient labelled data of a given clinical concept, which may not exist. On the other hand, our trie-based approach is reasonably fast and trivial to extend. We find that it is suitable for our purposes.

E.1.4 Bucketization of Triage Vitals

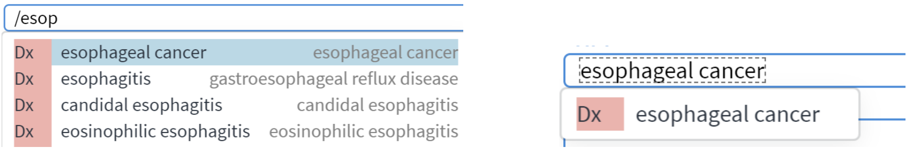
As described in Section 8.3.4, our best model for predicting a ranked list of relevant symptoms to document relied on a categorical featurization of triage vitals. The model simply uses the empiric frequencies of symptoms documented in a note, conditioned on the chief complaint c and a categorical representation $b(v)$ of the most abnormal vital v. We used medical guidelines to determine cutoffs for each vital as follows:

- *Temperature:* Temperatures above 100.4° are considered `HIGH` as they are medical-grade fevers. Temperatures below 97° are considered `LOW` as they are hypothermic. Otherwise, a temperature is considered `NORMAL`.
- *Respiratory rate:* A respiratory rate above 20 breaths per minute is considered `HIGH`, as per [57]. A respiratory rate below 12 breaths per minute is considered `LOW`. Otherwise, the respiratory rate is considered `NORMAL`.
- *Blood oxygen level*: A pulse oximeter reading below 95% is considered `LOW` as per Mayo Clinic guidelines. Otherwise, the reading is considered `NORMAL`.
- *Heart rate:* A heart rate above 100 beats per minute (bpm) is considered `TACHYCARDIC`. A heart rate below 60 is considered `BRADYCARDIC`. Otherwise, it is considered `NORMAL`.
- *Blood pressure:* Based on guidelines set by the American Heart Association, a systolic BP under 120 mmHg and a diastolic BP under 80 mmHg constitutes a `NORMAL` BP. If the diastolic BP is under 80 mmHg but the systolic BP is between 120-130 mmHg, it is considered `ELEVATED` blood pressure. If the systolic BP is under 140 mmHg and the diastolic blood pressure is under 90 mmHg, this is characterized as `STAGE 1 HYPERTENSION`. Otherwise, if either reading is higher, it is `STAGE 2 HYPERTENSION`.
- *Age:* Based on the age distribution of patients in the hospital, we bucketized patients into six groups: `CHILD` (e.g., below 18), `18-33`, `34-48`, `48-64`, `64-77`, and `78+`.

E.2 Extended Autocomplete Performance

E.2.1 Autocompletion Scope and Type Detection

Here, we provide an algorithm sketch of our autocompletion scope and type detection framework. The algorithm greedily uses keywords that act as autocompletion triggers,


/esop
Dx esophageal cancer esophageal cancer
Dx esophagitis gastroesophageal reflux disease
Dx candidal esophagitis candidal esophagitis
Dx eosinophilic esophagitis eosinophilic esophagitis
esophageal cancer
Dx esophageal cancer

(a) Manual autocompletion trigger

(b) Retroactive tagging

Figure E-2: Screenshots of our backup data capture strategies in the case that autocompletion scope detection algorithms fail. (a) Users can insert a slash character (`/`), which acts as a manual trigger to force autocompletion. (b) Users can retroactively accept tags for candidate concepts that they typed but did not autocomplete.

and is run and updated as a physician types a clinical note. First, we initialize the scope and type of our autocompletion to be null. Then, for each word w in the text, we update the scope accordingly:

- If w is part of a autocompletion trigger phrase such as `presents with`, we turn the autocompletion scope on and suggest terms to the user. We set the autocompletion type based on the trigger (`presents with` maps to `SYMPTOM`.)
- If w is a continuation token such as `and`, `or`, or `,`, we maintain the current scope and autocompletion type.
- If w is part of a tagged concept c, we turn the autocompletion scope on, and set the autocompletion type to the concept type of c.
- Otherwise, w is treated as a stopword, in which case the autocompletion scope is turned off.

With this framework, the autocompletion scope and type is greedily set using a simple parsing algorithm that is rerun as the user types a new word.

E.2.2 Training Contextual Model for Conditions

Our contextual model to predict a ranking over conditions is a dual-branched network that takes in two inputs:

1. A Term Frequency-Inverse Document Frequency (TF-IDF) representation of the triage text using unigrams and bigrams. Vocabulary size is close to 22,000.

2. The binary presence of different *model relevancy buckets* (as defined in Section 8.3.2) in the patient's prior medical history. This is a length-227 binary vector.

These two inputs are both passed through two separate dense layers with ReLU activation, concatenated and passed through another dense layer, and then finally passed through element-wise sigmoid activations to generate probabilities per class. We train this model with stochastic gradient descent using a cross entropy loss function.

E.2.3 Retrospective Autocompletion Performance using MRR, MAP, and Keystroke Burden

From an information retrieval perspective, we can analyze the quality of our ranked list of suggested clinical concepts by using two standard metrics: the *mean reciprocal rank* (MRR) and *mean average precision* (MAP). Consider an ordered ranking $R = \{r_1, r_2, r_3, \cdots\}$ of suggested terms and a ground truth set of terms that the clinician wants to documented denoted by $T = \{r_{\pi(1)}, r_{\pi(2)}, r_{\pi(3)}, \cdots\}$. We define the MRR of these suggestions as

$$MRR = \frac{1}{|T|} \sum_{\{r_i \in R | r_i \in T\}} (\max(1, i - |T|))^{-1}$$

In other words, this measures the average excess rank of the suggested terms that actually occur in the ground-truth terms the clinician wants to document. An MRR of 1 indicates that k desired terms were in the top k suggestions. The MAP score, in contrast, measures the average proportion of ground-truth terms that occur in the top k suggested terms as k varies:

$$MAP = \frac{1}{|T|} \sum_{k=1}^{|T|} \text{AveP}(k)$$

where AveP(k) represents average precision of the top k suggested terms. A MAP of 1 indicates perfect precision.

Below, we compare the various models we prototyped to predict each clinical

Model Type	Keystroke Burden ↓	MAP ↑
Conditions		
Frequency-based baseline	3.44 ±0.09	0.08 ±0.01
One vs. Rest Logistic Regression on triage text T	3.02 ±0.09	0.08 ±0.02
OvR LR on T, EHR	2.81 ±0.08	0.15 ±0.02
Augmented OvR LR on T, EHR	2.71 ±0.08	0.16 ±0.01
Dual-branched neural network	2.57 ±0.07	0.27 ±0.02
Symptoms		
Empirical Conditioning on Chief Complaint	2.19±0.04	0.41 ±0.01
Empirical Conditioning on Chief Complaint, Vital	**2.09** ±0.03	0.44 ±0.01
One vs. Rest Logistic Regression	2.74±0.02	0.16 ±0.01
One vs. Rest Naive Bayes	2.51 ±0.03	0.30 ±0.01
Labs (ranked by frequency)	0.092 ±0.03	0.39 ±0.01
Medications (ranked by frequency)	3.28 ±0.04	0.03 ±0.01
Overall with autocomplete scope/type detection	3.13 ±0.05	0.27 ±0.06

Table E.2: Retrospective Evaluation of Keystroke Burden and MAP using Contextual Autocompletion. We report the mean keystroke burden/MAP for the contextual autocomplete models we prototyped for each concept type, following the conventions of Figure 8.1

concept type with MAP and keystroke burden. Results in terms of MRR are in Figure 8.1.

E.2.4 Ontologies and Code

The codebase for our analyses as well as our publicly-available ontologies for conditions, symptoms, labs, and medications can be found here: `https://github.com/clinicalml/ContextualAutocomplete_MLHC2020`.

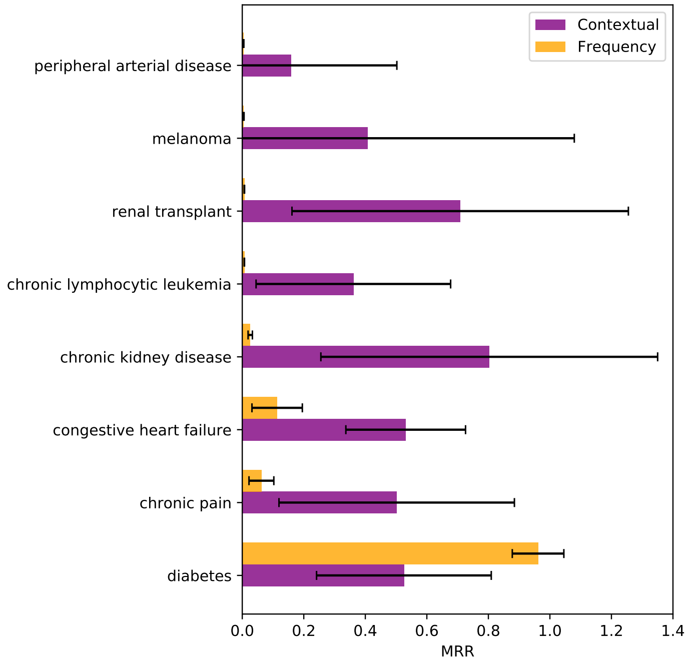


Figure E-3: Mean MRR for five conditions (± 95% CI from mean) using contextual and frequency-based autocompletion. Concepts were chosen to get representative samples of the data.

Bibliography

[1] Griffin Adams, Mert Ketenci, Shreyas Bhave, Adler J. Perotte, and Noémie Elhadad. Zero-shot clinical acronym expansion via latent meaning cells. In Emily Alsentzer, Matthew B. A. McDermott, Fabian Falck, Suproteem K. Sarkar, Subhrajit Roy, and Stephanie L. Hyland, editors, *Machine Learning for Health Workshop, ML4H@NeurIPS 2020, Virtual Event, 11 December 2020*, volume 136 of *Proceedings of Machine Learning Research*, pages 12–40. PMLR, 2020. URL `http://proceedings.mlr.press/v136/adams20a.html`.

[2] Ankur Agrawal and Gai Elhanan. Contrasting lexical similarity and formal definitions in SNOMED CT: Consistency and implications. *Journal of Biomedical Informatics*, 2014. ISSN 15320464. doi: 10.1016/j.jbi.2013.11.003.

[3] Monica Agrawal, Griffin Adams, Nathan Nussbaum, and Benjamin Birnbaum. Tifti: A framework for extracting drug intervals from longitudinal clinic notes. *arXiv preprint arXiv:1811.12793*, 2018.

[4] Monica Agrawal, Chloe O'Connell, Yasmin Fatemi, Ariel Levy, and David Sontag. Robust Benchmarking for Clinical Entity Recognition and Normalization. *Conference on Machine Learning for Healthcare (MLHC)*, 2020.

[5] Monica Agrawal, Stefan Hegselmann, Hunter Lang, Yoon Kim, and David Sontag. Large language models are few-shot clinical information extractors. *arXiv preprint arXiv:2205.12689*, 2022.

[6] Monica N Agrawal, Hunter Lang, Michael Offin, Lior Gazit, and David Sontag. Leveraging time irreversibility with order-contrastive pre-training. In *International Conference on Artificial Intelligence and Statistics*, pages 2330–2353. PMLR, 2022.

[7] Adil Ahmed, Subhash Chandra, Vitaly Herasevich, Ognjen Gajic, and Brian W. Pickering. The effect of two different electronic health record user interfaces on intensive care provider task load, errors of cognition, and performance. *Critical Care Medicine*, 39(7):1626–1634, July 2011. ISSN 1530-0293. doi: 10.1097/CCM.0b013e31821858a0.

[8] A. V. Aho and M.J. Corasick. Efficient string matching: An aid to bibliographic search. *Commun. ACM*, 18(6):333–340, June 1975. ISSN 0001-0782. doi: 10.1145/360825.360855. URL `https://doi.org/10.1145/360825.360855`.

[9] Alfred V. Aho and Margaret J. Corasick. Efficient String Matching: An Aid to Bibliographic Search. *Communications of the ACM*, 1975. ISSN 15577317. doi: 10.1145/360825.360855.

[10] Ilseyar Alimova and Elena Tutubalina. Multiple features for clinical relation extraction: a machine learning approach. *Journal of Biomedical Informatics*, 103:103382, 02 2020. doi: 10.1016/j.jbi.2020.103382.

[11] D. Aljabri, A. Dumitrascu, C. Burton, L. White, M. Khan, S. Xirasagar, R. Horner, and J. Naessens. Patient portal adoption and use by hospitalized cancer patients: a retrospective study of its impact on adverse events, utilization, and patient satisfaction. *BMC Medical Informatics and Decision Making*, 18(1):70, 2018. ISSN 1472-6947. doi: 10.1186/s12911-018-0644-4. URL `https://doi.org/10.1186/s12911-018-0644-4`.

[12] J. J. Allison, T. C. Wall, C. M. Spettell, J. Calhoun, C. A. Fargason, R. W. Kobylinski, R. Farmer, and C. Kiefe. The art and science of chart review. *The Joint Commission journal on quality improvement*, 2000. ISSN 10703241. doi: 10.1016/S1070-3241(00)26009-4.

[13] Emily Alsentzer, John R Murphy, Willie Boag, Wei-Hung Weng, Di Jin, Tristan Naumann, and Matthew McDermott. Publicly available clinical bert embeddings. *arXiv preprint arXiv:1904.03323*, 2019.

[14] Waleed Ammar, Dirk Groeneveld, Chandra Bhagavatula, Iz Beltagy, Miles Crawford, Doug Downey, Jason Dunkelberger, Ahmed Elgohary, Sergey Feldman, Vu Ha, et al. Construction of the literature graph in semantic scholar. *arXiv preprint arXiv:1805.02262*, 2018.

[15] Alan R. Aronson and François Michel Lang. An overview of MetaMap: Historical perspective and recent advances. *Journal of the American Medical Informatics Association*, 2010. ISSN 10675027. doi: 10.1136/jamia.2009.002733.

[16] Jim Atherton. Development of the Electronic Health Record. *AMA Journal of Ethics*, 13(3):186–189, March 2011. ISSN 2376-6980. doi: 10.1001/virtualmentor.2011.13.3.mhst1-1103.

[17] Aristotle Bamias, Bernard Escudier, Cora N. Sternberg, Flora Zagouri, Athanasios Dellis, Bob Djavan, Kimon Tzannis, Loukas Kontovinis, Konstantinos Stravodimos, Athanasios Papatsoris, Dionysios Mitropoulos, Charalampos Deliveliotis, Meletios-Athanasios Dimopoulos, and Constantine A. Constantinides. Current Clinical Practice Guidelines for the Treatment of Renal Cell Carcinoma: A Systematic Review and Critical Evaluation. *The Oncologist*, 2017. ISSN 1083-7159. doi: 10.1634/theoncologist.2016-0435.

[18] Vin Shen Ban, Christopher J Madden, Travis Browning, Ellen O'Connell, Bradley F Marple, and Brett Moran. A novel use of the discrete templated notes within an electronic health record software to monitor resident supervision.

Journal of the American Medical Informatics Association, 24(e1):e2–e8, April 2017. ISSN 1067-5027, 1527-974X. doi: 10.1093/jamia/ocw078.

[19] Imon Banerjee, Selen Bozkurt, Jennifer Lee Caswell-Jin, Allison W. Kurian, and Daniel L. Rubin. Natural Language Processing Approaches to Detect the Timeline of Metastatic Recurrence of Breast Cancer. *JCO Clinical Cancer Informatics*, 2019. ISSN 2473-4276. doi: 10.1200/cci.19.00034.

[20] Aaron Bangor, Philip T. Kortum, and James T. Miller. An Empirical Evaluation of the System Usability Scale. *International Journal of Human-Computer Interaction*, 24(6):574–594, July 2008. ISSN 1044-7318, 1532-7590. doi: 10.1080/10447310802205776.

[21] Gagan Bansal, Besmira Nushi, Ece Kamar, Walter S Lasecki, Daniel S Weld, and Eric Horvitz. Beyond Accuracy: The Role of Mental Models in Human-AI Team Performance. *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*, 2019.

[22] Hubert Banville, Omar Chehab, Aapo Hyvärinen, Denis-Alexander Engemann, and Alexandre Gramfort. Uncovering the structure of clinical eeg signals with self-supervised learning. *Journal of Neural Engineering*, 18(4):046020, 2021.

[23] Hilda Bastian, Paul Glasziou, and Iain Chalmers. Seventy-five trials and eleven systematic reviews a day: how will we ever keep up? *PLoS medicine*, 7(9): e1000326, 2010.

[24] David W. Bates and Atul A. Gawande. Improving Safety with Information Technology. *New England Journal of Medicine*, 348(25):2526–2534, June 2003. ISSN 0028-4793. doi: 10.1056/NEJMsa020847.

[25] Sigall K Bell, Tom Delbanco, Joann G Elmore, Patricia S Fitzgerald, Alan Fossa, Kendall Harcourt, Suzanne G Leveille, Thomas H Payne, Rebecca A Stametz, Jan Walker, et al. Frequency and types of patient-reported errors in electronic health record ambulatory care notes. *JAMA network open*, 3(6): e205867–e205867, 2020.

[26] Marc L. Berger, Melissa D. Curtis, Gregory Smith, James Harnett, and Amy P. Abernethy. Opportunities and challenges in leveraging electronic health record data in oncology, 5 2016. ISSN 17448301. URL `https://www.futuremedicine.com/doi/abs/10.2217/fon-2015-0043`.

[27] Dorrit Billman and Eric A. Bier. Medical sensemaking with entity workspace. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '07, pages 229–232, New York, NY, USA, April 2007. Association for Computing Machinery. ISBN 978-1-59593-593-9. doi: 10.1145/1240624.1240662.

[28] Benjamin Birnbaum, Nathan Nussbaum, Katharina Seidl-Rathkopf, Monica Agrawal, Melissa Estevez, Evan Estola, Joshua Haimson, Lucy He, Peter Larson,

and Paul Richardson. Model-assisted cohort selection with bias analysis for generating large-scale cohorts from the ehr for oncology research. *arXiv preprint arXiv:2001.09765*, 2020.

[29] Joanne Bleackley and So Youn Rachel Kim. The merit and agony of retrospective chart reviews: A medical student's perspective. *British Columbia Medical Journal*, 55:374–375, 2013.

[30] Olivier Bodenreider. The Unified Medical Language System (UMLS): Integrating biomedical terminology. *Nucleic Acids Research*, 2004. ISSN 03051048. doi: 10.1093/nar/gkh061.

[31] Olivier Bodenreider. The unified medical language system (umls): integrating biomedical terminology. *Nucleic acids research*, 32(suppl_1):D267–D270, 2004.

[32] E. Bolton, D. Hall, M. Yasunaga, T. Lee, C. Manning, and P. Liang. Stanford crfm introduces pubmedgpt 2.7b. *Stanford Human-Centered Artificial Intelligence*, 2022.

[33] Matthew Brehmer, Stephen Ingram, Jonathan Stray, and Tamara Munzner. Overview: The design, adoption, and analysis of a visual document mining tool for investigative journalists. *IEEE Transactions on Visualization and Computer Graphics*, 2014. ISSN 10772626. doi: 10.1109/TVCG.2014.2346431.

[34] Eugene M. Breydo, Julia T. Chu, and Alexander Turchin. Identification of inactive medications in narrative medical text. *AMIA ... Annual Symposium proceedings / AMIA Symposium. AMIA Symposium*, 2008. ISSN 1942597X.

[35] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[36] Joel Buchanan. Accelerating the Benefits of the Problem Oriented Medical Record. *Applied Clinical Informatics*, 8(1):180–190, February 2017. ISSN 1869-0327. doi: 10.4338/ACI-2016-04-IE-0054.

[37] Zana Buçinca, Phoebe Lin, Krzysztof Z. Gajos, and Elena L. Glassman. Proxy tasks and subjective measures can be misleading in evaluating explainable AI systems. In *International Conference on Intelligent User Interfaces, Proceedings IUI*, 2020. ISBN 9781450371186. doi: 10.1145/3377325.3377498.

[38] Carrie J. Cai, Samantha Winter, David Steiner, Lauren Wilcox, and Michael Terry. "Hello Ai": Uncovering the onboarding needs of medical practitioners for human–AI collaborative decision-making, 2019. ISSN 25730142.

[39] A. Calvitti, N. Farber, Y. Chen, D. Zuest, L. Liu, K. Bell, B. Gray, and Z. Agha. Temporal Analysis of Physicians' EHR Workflow during Outpatient Visits. In

2012 IEEE Second International Conference on Healthcare Informatics, Imaging and Systems Biology, pages 140–140, September 2012. doi: 10.1109/HISB.2012.65.

[40] J A Cannon-Bowers, E Salas, and S Converse. Shared mental models in expert team decision making. *Individual and Group Decision Making*, 1993.

[41] P. Carayon, T. Wetterneck, B. Alyousef, R. Brown, R. Cartmill, K. McGuire, P.L.T. Hoonakker, J. Slagle, K. Roy, J. Walker, M. Weinger, A. Xie, and K. Wood. Impact of electronic health record technology on the work and workflow of physicians in the intensive care unit. *International Journal of Medical Informatics*, 84, 04 2015. doi: 10.1016/j.ijmedinf.2015.04.002.

[42] David S. Carrell, Scott Halgrim, Diem-Thy Tran, Diana S. M. Buist, Jessica Chubak, Wendy W. Chapman, and Guergana Savova. Using Natural Language Processing to Improve Efficiency of Manual Chart Abstraction in Research: The Case of Breast Cancer Recurrence. *American Journal of Epidemiology*, 179(6):749–758, 3 2014. ISSN 1476-6256. doi: 10.1093/aje/kwt441. URL `https://academic.oup.com/aje/article-lookup/doi/10.1093/aje/kwt441`.

[43] Alison Cave, Xavier Kurz, and Peter Arlett. Real-World Data for Regulatory Decision Making: Challenges and Possible Solutions for Europe, 2019. ISSN 15326535.

[44] Centers for Medicare and Medicaid Services. Evaluation and management services guide, 2020.

[45] W. W. Chapman, W. Bridewell, P. Hanbury, G. F. Cooper, and B. G. Buchanan. A simple algorithm for identifying negated findings and diseases in discharge summaries. *Journal of Biomedical Informatics*, 34(5):301–310, Oct 2001.

[46] Wendy W. Chapman, Will Bridewell, Paul Hanbury, Gregory F. Cooper, and Bruce G. Buchanan. A Simple Algorithm for Identifying Negated Findings and Diseases in Discharge Summaries. *Journal of Biomedical Informatics*, 34(5): 301–310, October 2001. ISSN 1532-0464. doi: 10.1006/jbin.2001.1029.

[47] Wendy W. Chapman, Prakash M. Nadkarni, Lynette Hirschman, Leonard W. D'Avolio, Guergana K. Savova, and Ozlem Uzuner. Overcoming barriers to NLP for clinical text: The role of shared tasks and the need for additional creative solutions, 2011. ISSN 10675027.

[48] Irene Y Chen, Emily Alsentzer, Hyesun Park, Richard Thomas, Babina Gosangi, Rahul Gujrathi, and Bharti Khurana. Intimate partner violence and injury prediction from radiology reports. In *BIOCOMPUTING 2021: Proceedings of the Pacific Symposium*, pages 55–66. World Scientific, 2020.

[49] Mia Xu Chen, Benjamin N. Lee, Gagan Bansal, Yuan Cao, Shuyuan Zhang, Justin Lu, Jackie Tsay, Yinan Wang, Andrew M. Dai, Zhifeng Chen, Timothy

Sohn, and Yonghui Wu. Gmail smart compose: Real-time assisted writing. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '19, page 2287–2295, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450362016. doi: 10.1145/3292500.3330723. URL `https://doi.org/10.1145/3292500.3330723`.

[50] Yan Chen, Huanying (Helen) Gu, Yehoshua Perl, and James Geller. Structural group-based auditing of missing hierarchical relationships in UMLS. *Journal of Biomedical Informatics*, 2009. ISSN 15320464. doi: 10.1016/j.jbi.2008.08.006.

[51] Asad J Choudhry, Yaser MK Baghdadi, Amy E Wagie, Elizabeth B Habermann, Stephanie F Heller, Donald H Jenkins, Daniel C Cullinane, and Martin D Zielinski. Readability of discharge summaries: with what level of information are we dismissing our patients? *The American Journal of Surgery*, 211(3):631–636, 2016.

[52] Ching-Yao Chuang, Joshua Robinson, Yen-Chen Lin, Antonio Torralba, and Stefanie Jegelka. Debiased contrastive learning. In *NeurIPS*, 2020.

[53] J. J. Cimino, H. Min, and Y. Perl. Consistency across the hierarchies of the UMLS Semantic Network and Metathesaurus. *Journal of Biomedical Informatics*, 2003. ISSN 15320464. doi: 10.1016/j.jbi.2003.11.001.

[54] Google Cloud. Healthcare Natural Language API | Cloud Healthcare API | Google Cloud. https://cloud.google.com/healthcare/docs/concepts/nlp, July 2021.

[55] CMS. 1997 Documentation Guidelines for Evaluation and Management Services.

[56] Enrico Coiera. When Conversation Is Better Than Computation. *Journal of the American Medical Informatics Association : JAMIA*, 7(3):277–286, 2000. ISSN 1067-5027.

[57] M. A. Cretikos, R. Bellomo, K. Hillman, J. Chen, S. Finfer, and A. Flabouris. Respiratory rate: the neglected vital sign. *Med. J. Aust.*, 188(11):657–659, Jun 2008.

[58] Leyang Cui, Yu Wu, Jian Liu, Sen Yang, and Yue Zhang. Template-based named entity recognition using bart. *arXiv preprint arXiv:2106.01760*, 2021.

[59] Steven J. Davidson, Frank L. Zwemer, Larry A. Nathanson, Kenneth N. Sable, and Abu N.G.A. Khan. Where's the beef? The promise and the reality of clinical documentation. In *Academic Emergency Medicine*, 2004. doi: 10.1197/j.aem.2004.08.004.

[60] Dina Demner-Fushman, Wendy W. Chapman, and Clement J. McDonald. What can natural language processing do for clinical decision support?, 2009. ISSN 15320464.

[61] J. Devlin, M. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL `https://www.aclweb.org/anthology/N19-1423`.

[62] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[63] Nathaniel Diamant, Erik Reinertsen, Steven Song, Aaron D Aguirre, Collin M Stultz, and Puneet Batra. Patient contrastive learning: A performant, expressive, and practical approach to electrocardiogram modeling. *PLOS Computational Biology*, 18(2):e1009862, 2022.

[64] L. Doré, M. Lavril, F. C. Jean, and P. Degoulet. An object oriented computer-based patient record reference model. *Proceedings of the Annual Symposium on Computer Application in Medical Care*, pages 377–381, 1995. ISSN 0195-4210.

[65] Wlodzislaw Duch, Rudy Setiono, and Jacek M Zurada. Computational intelligence methods for rule-based data understanding. *Proceedings of the IEEE*, 92 (5):771–805, 2004.

[66] Noémie Elhadad, Sameer Pradhan, Sharon Gorman, Suresh Manandhar, Wendy Chapman, and Guergana Savova. SemEval-2015 Task 14: Analysis of Clinical Text. 2015. doi: 10.18653/v1/s15-2051.

[67] Meng Fang, Yuan Li, and Trevor Cohn. Learning how to active learn: A deep reinforcement learning approach. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 595–605, 2017.

[68] O. Farri, A. Rahman, K.A. Monsen, R. Zhang, S.V. Pakhomov, D.S. Pieczkiewicz, S.M. Speedie, and G.B. Melton. Impact of a Prototype Visualization Tool for New Information in EHR Clinical Documents. *Applied Clinical Informatics*, 3(4): 404–418, October 2012. ISSN 1869-0327. doi: 10.4338/ACI-2012-05-RA-0017.

[69] FDA. Framework for FDA's Real-World Evidence Program. pages 1–37, 2018. ISSN 00070610. doi: 10.1038/sj.bdj.2012.1047.

[70] Basura Fernando, Efstratios Gavves, Jose M Oramas, Amir Ghodrati, and Tinne Tuytelaars. Modeling video evolution for action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5378–5387, 2015.

[71] Elisa Ferracane, Iain Marshall, Byron C Wallace, and Katrin Erk. Leveraging coreference to identify arms in medical abstracts: An experimental study. In

Proceedings of the Seventh International Workshop on Health Text Mining and Information Analysis, pages 86–95, 2016.

[72] Paolo Ferragina and Ugo Scaiella. TAGME: On-the-fly annotation of short text fragments (by Wikipedia entities). In *International Conference on Information and Knowledge Management, Proceedings*, 2010. ISBN 9781450300995. doi: 10.1145/1871437.1871689.

[73] Karën Fort and Benoît Sagot. Influence of pre-annotation on POS-tagged corpus development. In *ACL 2010 - LAW 2010: 4th Linguistic Annotation Workshop, Proceedings*, 2010. ISBN 1932432728.

[74] Ian Fox, Harry Rubin-Falcone, and Jenna Wiens. Learning through limited self-supervision: Improving time-series classification without additional data via auxiliary tasks. 2019.

[75] Jean-Yves Franceschi, Aymeric Dieuleveut, and Martin Jaggi. Unsupervised scalable representation learning for multivariate time series. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

[76] Jessica M. Franklin and Sebastian Schneeweiss. When and How Can Real World Data Analyses Substitute for Randomized Controlled Trials? *Clinical Pharmacology and Therapeutics*, 2017. ISSN 15326535. doi: 10.1002/cpt.857.

[77] Mark W. Friedberg, Peggy G. Chen, Kristin R. Van Busum, Frances Aunon, Chau Pham, John Caloyeras, Soeren Mattke, Emma Pitchforth, Denise D. Quigley, Robert H. Brook, F. Jay Crosson, and Michael Tutty. Factors Affecting Physician Professional Satisfaction and Their Implications for Patient Care, Health Systems, and Health Policy. *Rand Health Quarterly*, 3(4), December 2014. ISSN 2162-8254.

[78] Thomas R. Frieden. Evidence for health decision making-beyond randomized, controlled trials, 2017. ISSN 15334406.

[79] Carol Friedman. Medlee-a medical language extraction and encoding system. *Columbia University, and Queens College of CUNY*, 1995.

[80] Carol Friedman, Lyudmila Shagina, Yves Lussier, and George Hripcsak. Automated encoding of clinical documents based on natural language processing. *Journal of the American Medical Informatics Association*, 2004. ISSN 10675027. doi: 10.1197/jamia.M1552.

[81] Kuzman Ganchev, Fernando Pereira, Mark Mandel, Steven Carroll, and Peter White. Semi-automated named entity annotation. In *ACL 2007: The LAW - Proceedings of The Linguistic Annotation Workshop*, 2007. doi: 10.3115/1642059.1642068.

[82] R. L. Gardner, E. Cooper, J. Haskell, D. A. Harris, S. Poplau, P. J. Kroth, and M. Linzer. Physician stress and burnout: the impact of health information technology. *Journal of American Medical Informatics Association*, 26(2):106–114, Feb 2019.

[83] Yonatan Geifman and Ran El-Yaniv. SelectiveNet: A deep neural network with an integrated reject option. In *36th International Conference on Machine Learning, ICML 2019*, 2019. ISBN 9781510886988.

[84] M. Gerard, H. Chimowitz, A. Fossa, F. Bourgeois, L Fernandez, and SK Bell. The importance of visit notes on patient portals for engaging less educated or nonwhite patients: Survey study. *J Med Internet Res*, 20(5):e191, May 2018. ISSN 1438-8871. doi: 10.2196/jmir.9196. URL `http://www.jmir.org/2018/5/e191/`.

[85] Katy Ilonka Gero, Zahra Ashktorab, Casey Dugan, Qian Pan, James Johnson, Werner Geyer, Maria Ruiz, Sarah Miller, David R. Millen, Murray Campbell, Sadhana Kumaravel, and Wei Zhang. Mental Models of AI Agents in a Cooperative Game Setting. 2020. doi: 10.1145/3313831.3376316.

[86] Bryan Gibson, Jorie Butler, Maryan Zirkle, Kenric Hammond, and Charlene Weir. Foraging for Information in the EHR: The Search for Adherence Related Information by Mental Health Clinicians. *AMIA Annual Symposium Proceedings*, 2016:600–608, February 2017. ISSN 1942-597X.

[87] Glenn T. Gobbel Dr, Jennifer Garvin, Ruth Reeves, Robert M. Cronin, Julia Heavirland, Jenifer Williams, Allison Weaver, Shrimalini Jayaramaraja, Dario Giuse, Theodore Speroff, Steven H. Brown, Hua Xu, and Michael E. Matheny. Assisted annotation of medical free text using RapTAT. *Journal of the American Medical Informatics Association*, 2014. ISSN 1527974X. doi: 10.1136/amiajnl-2013-002255.

[88] Jen J Gong, Tristan Naumann, Peter Szolovits, and John V Guttag. Predicting clinical outcomes across changing electronic health record systems. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1497–1505, 2017.

[89] Divya Gopinath. Meng thesis: Ml-driven clinical documentation. Master's thesis, Massachusetts Institute of Technology, Department of Electrical Engineering and Computer Science, 2020.

[90] Divya Gopinath, Monica Agrawal, Luke Murray, Steven Horng, David Karger, and David Sontag. Fast, Structured Clinical Documentation via Contextual Autocomplete. *arXiv:2007.15153 [cs, stat]*, July 2020.

[91] Carsten Görg, Zhicheng Liu, Jaeyeon Kihm, Jaegul Choo, Haesun Park, and John Stasko. Combining computational analyses and interactive visualization for document exploration and sensemaking in jigsaw. *IEEE Transactions on*

Visualization and Computer Graphics, 2013. ISSN 10772626. doi: 10.1109/TVCG.2012.324.

[92] Ross Goroshin, Joan Bruna, Jonathan Tompson, David Eigen, and Yann LeCun. Unsupervised learning of spatiotemporally coherent metrics. In *Proceedings of the IEEE international conference on computer vision*, pages 4086–4093, 2015.

[93] Spence Green, Jason Chuang, Jeffrey Heer, and Christopher D. Manning. Predictive translation memory: A mixed-initiative system for human language translation. In *UIST 2014 - Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology*, 2014. ISBN 9781450330695. doi: 10.1145/2642918.2647408.

[94] N.R. Greenbaum, Y. Jernite, Y. Halpern, S. Calder, L.A. Nathanson, D. Sontag, and S. Horng. Contextual autocomplete: A novel user interface using machine learning to improve ontology usage and structured data capture for presenting problems in the emergency department. *bioRxiv*, 2017. doi: 10.1101/127092. URL `https://www.biorxiv.org/content/early/2017/04/12/127092`.

[95] Barbara J. Grosz and Sarit Kraus. The Evolution of Sharedplans. 1999. doi: 10.1007/978-94-015-9204-8_10.

[96] Yu Gu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. Domain-specific language model pretraining for biomedical natural language processing. *ACM Transactions on Computing for Healthcare (HEALTH)*, 3(1):1–23, 2021.

[97] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *34th International Conference on Machine Learning, ICML 2017*, 2017. ISBN 9781510855144.

[98] Bernal Jiménez Gutiérrez, Nikolas McNeal, Clay Washington, You Chen, Lang Li, Huan Sun, and Yu Su. Thinking about gpt-3 in-context learning for biomedical ie? think again. *arXiv preprint arXiv:2203.08410*, 2022.

[99] Benedict Guzman, Isabel Metzger, Yindalon Aphinyanaphongs, Himanshu Grover, et al. Assessment of amazon comprehend medical: Medication information extraction. *arXiv preprint arXiv:2002.00481*, 2020.

[100] Janet Haas, Suzanne Bakken, Tiffani J. Bright, Genevieve B. Melton, Peter Stetson, and Stephen B. Johnson. Clinicians' perceptions of usability of eNote. *AMIA ... Annual Symposium proceedings. AMIA Symposium*, page 973, 2005. ISSN 1942-597X.

[101] Janet P. Haas, Suzanne Bakken, Tiffani J. Bright, Genevieve B. Melton, Peter Stetson, and Stephen B. Johnson. Clinicians' Perceptions of Usability of eNote. *AMIA Annual Symposium Proceedings*, 2005:973, 2005. ISSN 1942-597X.

[102] Michael Halper, C. Paul Morrey, Yan Chen, Gai Elhanan, George Hripcsak, and Yehoshua Perl. Auditing hierarchical cycles to locate other inconsistencies in the UMLS. *AMIA ... Annual Symposium proceedings / AMIA Symposium. AMIA Symposium*, 2011. ISSN 1942597X.

[103] A. M. Harbourt, E. J. Syed, W. T. Hole, and L. C. Kingsland. The ranking algorithm of the Coach browser for the UMLS metathesaurus. *Proceedings / the ... Annual Symposium on Computer Application [sic] in Medical Care. Symposium on Computer Applications in Medical Care*, 1993. ISSN 01954210.

[104] Michael R. Harrison, Bradford R. Hirsch, Daniel J. George, Mark S. Walker, Connie Chen, Beata Korytowsky, Edward Stepanski, and Amy P. Abernethy. Real-World Outcomes in Metastatic Renal Cell Carcinoma: Insights From a Joint Community-Academic Registry. *Journal of Oncology Practice*, 2014. ISSN 1554-7477. doi: 10.1200/jop.2013.001180.

[105] Pamela Hartzband and Jerome Groopman. Off the record–avoiding the pitfalls of going electronic. *The New England Journal of Medicine*, 358(16):1656–1658, April 2008. ISSN 1533-4406. doi: 10.1056/NEJMp0802221.

[106] Jeffrey Heer. Agency plus automation: Designing artificial intelligence into interactive systems. *Proceedings of the National Academy of Sciences of the United States of America*, 2019. ISSN 10916490. doi: 10.1073/pnas.1807184115.

[107] Sam Henry, Kevin Buchan, Michele Filannino, Amber Stubbs, and Ozlem Uzuner. 2018 n2c2 shared task on adverse drug events and medication extraction in electronic health records. *Journal of the American Medical Informatics Association*, 27(1):3–12, 2020.

[108] Sebastian Hergeth, Lutz Lorenz, Roman Vilimek, and Josef F. Krems. Keep Your Scanners Peeled: Gaze Behavior as a Measure of Automation Trust during Highly Automated Driving. *Human Factors*, 2016. ISSN 15478181. doi: 10.1177/0018720815625744.

[109] Bradford R. Hirsch, Michael R. Harrison, Daniel J. George, Mark S. Walker, Connie Chen, Beata Korytowsky, Edward Stepanski, and Amy P. Abernethy. Use of "Real-World" data to describe adverse events during the treatment of metastatic renal cell carcinoma in routine clinical practice. *Medical Oncology*, 2014. ISSN 1559131X. doi: 10.1007/s12032-014-0156-8.

[110] Jamie S Hirsch, Jessica S Tanenbaum, Sharon Lipsky Gorman, Connie Liu, Eric Schmitz, Dritan Hashorva, Artem Ervits, David Vawdrey, Marc Sturm, and Noémie Elhadad. HARVEST, a longitudinal patient record summarizer. *Journal of the American Medical Informatics Association*, 22(2):263–274, March 2015. ISSN 1067-5027. doi: 10.1136/amiajnl-2014-002945.

[111] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, November 1997. ISSN 0899-7667. doi: 10.1162/neco.1997.9.8.1735. URL `https://doi.org/10.1162/neco.1997.9.8.1735`.

[112] Johannes Hoffart, Mohamed Amir Yosef, Ilaria Bordino, Hagen Fürstenau, Manfred Pinkal, Marc Spaniol, Bilyana Taneva, Stefan Thater, and Gerhard Weikum. Robust disambiguation of named entities in text. In *EMNLP 2011 - Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, 2011. ISBN 1937284115.

[113] Andreas Holzinger. Interactive machine learning for health informatics: when do we need the human-in-the-loop? *Brain Informatics*, 2016. ISSN 21984026. doi: 10.1007/s40708-016-0042-6.

[114] K. Huang, J. Altosaar, and R. Ranganath. Clinicalbert: Modeling clinical notes and predicting hospital readmission. *CoRR*, abs/1904.05342, 2019. URL `http://arxiv.org/abs/1904.05342`.

[115] Kexin Huang, Jaan Altosaar, and Rajesh Ranganath. Clinicalbert: Modeling clinical notes and predicting hospital readmission. *arXiv preprint arXiv:1904.05342*, 2019.

[116] Aapo Hyvärinen and Hiroshi Morioka. Nonlinear ica of temporally dependent stationary sources. In *Artificial Intelligence and Statistics*, pages 460–469. PMLR, 2017.

[117] Inc. Grammarly. Grammarly: Free Writing Assistant, 2018.

[118] Peter B. Jensen, Lars J. Jensen, and Soøren Brunak. Mining electronic health records: Towards better research applications and clinical care, 2012. ISSN 14710056.

[119] Min Jiang, Yukun Chen, Mei Liu, S Trent Rosenbloom, Subramani Mani, Joshua C Denny, and Hua Xu. A study of machine-learning-based approaches to extract clinical entities and their assertions from discharge summaries. *Journal of the American Medical Informatics Association*, 18(5):601–606, 2011.

[120] Alistair E.W. Johnson, Tom J. Pollard, Lu Shen, Li Wei H. Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G. Mark. MIMIC-III, a freely accessible critical care database. *Scientific Data*, 2016. ISSN 20524463. doi: 10.1038/sdata.2016.35.

[121] Alistair EW Johnson, Tom J Pollard, Lu Shen, Li-wei H Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. Mimic-iii, a freely accessible critical care database. *Scientific data*, 3(1):1–9, 2016.

[122] Stephen B. Johnson, Suzanne Bakken, Daniel Dine, Sookyung Hyun, Eneida Mendonça, Frances Morrison, Tiffani Bright, Tielman Van Vleck, Jesse Wrenn, and Peter Stetson. An Electronic Health Record Based on Structured Narrative. *Journal of the American Medical Informatics Association*, 15(1):54–64, January 2008. ISSN 1067-5027. doi: 10.1197/jamia.M2131.

[123] Patrick W. Jordan, B. Thomas, Ian Lyall McClelland, and Bernard Weerdmeester. *Usability Evaluation In Industry*. CRC Press, June 1996. ISBN 978-1-4987-1041-1.

[124] Kenneth L Kehl, Haitham Elmarakeby, Mizuki Nishino, Eliezer M Van Allen, Eva M Lepisto, Michael J Hassett, Bruce E Johnson, and Deborah Schrag. Assessment of deep natural language processing in ascertaining oncologic outcomes from radiology reports. *JAMA oncology*, 5(10):1421–1429, 2019.

[125] Kenneth L Kehl, Wenxin Xu, Eva Lepisto, Haitham Elmarakeby, Michael J Hassett, Eliezer M Van Allen, Bruce E Johnson, and Deborah Schrag. Natural language processing to ascertain cancer outcomes from medical oncologist notes. *JCO Clinical Cancer Informatics*, 4:680–690, 2020.

[126] Tom Kenter and Diana Maynard. Using gate as an annotation tool. pages 1–11, 2004. URL `https://gate.ac.uk/sale/am/annotationmanual-gate2.pdf`.

[127] Vijay Keswani, Matthew Lease, and Krishnaram Kenthapadi. Towards unbiased and accurate deferral to multiple experts. In *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society*, pages 154–165, 2021.

[128] A.A. Khorana. Physician as typist. *Journal of Clinical Oncology*, 28(24):3899–3900, 2010. doi: 10.1200/JCO.2010.29.4504. URL `https://doi.org/10.1200/JCO.2010.29.4504`. PMID: 20547988.

[129] Sean Khozin, Gideon M. Blumenthal, and Richard Pazdur. Real-world Data for Clinical Evidence Generation in Oncology. *Journal of the National Cancer Institute*, 2017. ISSN 14602105. doi: 10.1093/jnci/djx187.

[130] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[131] Dani Kiyasseh, Tingting Zhu, and David A Clifton. Clocs: contrastive learning of cardiac signals across space, time, and patients. In *International Conference on Machine Learning*, pages 5606–5615. PMLR, 2021.

[132] Jon Kleinberg, Himabindu Lakkaraju, Jure Leskovec, Jens Ludwig, and Sendhil Mullainathan. Human decisions and machine predictions. *Quarterly Journal of Economics*, 2018. ISSN 15314650. doi: 10.1093/qje/qjx032.

[133] Susan Koch-Weser, William Dejong, and Rima E. Rudd. Medical word use in clinical encounters. *Health expectations : an international journal of public*

participation in health care and health policy, 12(4):371–382, Dec 2009. ISSN 1369-7625. doi: 10.1111/j.1369-7625.2009.00555.x. URL `https://pubmed.ncbi.nlm.nih.gov/19709316`. PMC5060502[pmcid].

[134] Nikolaos Kolitsas, Octavian Eugen Ganea, and Thomas Hofmann. End-to-end neural entity linking. In *CoNLL 2018 - 22nd Conference on Computational Natural Language Learning, Proceedings*, 2018. ISBN 9781948087728. doi: 10.18653/v1/k18-1050.

[135] Sandeep Konam and Shivdev Rao. Abridge: A mission driven approach to machine learning for healthcare conversation. *Journal of Commercial Biotechnology*, 26(2), 2021.

[136] Bevan Koopman, Guido Zuccon, Anthony Nguyen, Anton Bergheim, and Narelle Grayson. Extracting cancer mortality statistics from death certificates: A hybrid machine learning and rule-based approach for common and rare cancers. *Artificial intelligence in medicine*, 89:1–9, 2018.

[137] Kundan Krishna, Amy Pavel, Benjamin Schloss, Jeffrey P Bigham, and Zachary C Lipton. Extracting structured data from physician-patient conversations by predicting noteworthy utterances. In *Explainable AI in Healthcare and Medicine*, pages 155–169. Springer, 2021.

[138] Philip J Kroth, Nancy Morioka-Douglas, Sharry Veres, Katherine Pollock, Stewart Babbott, Sara Poplau, Katherine Corrigan, and Mark Linzer. The electronic elephant in the room: Physicians and the electronic health record. *JAMIA Open*, 2018. ISSN 2574-2531. doi: 10.1093/jamiaopen/ooy016.

[139] Ivy Fenton Kuhn. Abbreviations and acronyms in healthcare: when shorter isn't sweeter. *Pediatric nursing*, 33(5), 2007.

[140] Thomson Kuhn, Peter Basch, Michael Barr, and Thomas Yackel. Clinical Documentation in the 21st Century: Executive Summary of a Policy Position Paper From the American College of Physicians. *Annals of Internal Medicine*, 162(4):301–303, February 2015. ISSN 0003-4819. doi: 10.7326/M14-2128.

[141] Chinmay Kulkarni, Steven P. Dow, and Scott R. Klemmer. Early and repeated exposure to examples improves creative work. In *Design Thinking Research: Building Innovation Eco-Systems*. 2014. ISBN 9783319013039. doi: 10.1007/978-3-319-01303-9_4.

[142] Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. Neural architectures for named entity recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 260–270, San Diego, California, June 2016. Association for Computational Linguistics. doi: 10.18653/v1/N16-1030. URL `https://www.aclweb.org/anthology/N16-1030`.

[143] Hunter Lang, Monica Agrawal, Yoon Kim, and David Sontag. Co-training improves prompt-based learning for large language models. *arXiv preprint arXiv:2202.00828*, 2022.

[144] Hunter Lang, Aravindan Vijayaraghavan, and David Sontag. Training subset selection for weak supervision. *arXiv preprint arXiv:2206.02914*, 2022.

[145] Hsin-Ying Lee, Jia-Bin Huang, Maneesh Singh, and Ming-Hsuan Yang. Unsupervised representation learning by sorting sequences. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 667–676, 2017.

[146] Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240, 2020.

[147] John D. Lee and Katrina A. See. Trust in automation: Designing for appropriate reliance, 2004. ISSN 00187208.

[148] Kenton Lee, Luheng He, Mike Lewis, and Luke Zettlemoyer. End-to-end neural coreference resolution. *arXiv preprint arXiv:1707.07045*, 2017.

[149] Eric Lehman, Vladislav Lialin, Katelyn Y Legaspi, Anne Janelle R Sy, Patricia Therese S Pile, Nicole Rose I Alberto, Richard Raymund R Ragasa, Corinna Victoria M Puyat, Isabelle Rose I Alberto, Pia Gabrielle I Alfonso, et al. Learning to ask like a physician. *arXiv preprint arXiv:2206.02696*, 2022.

[150] Ariel Levy, Monica Agrawal, Arvind Satyanarayan, and David Sontag. Assessing the impact of automated suggestions on decision making: Domain experts mediate model errors but take less initiative. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, pages 1–13, 2021.

[151] Brenna Li, Noah Crampton, Thomas Yeates, Yu Xia, Xirong Tian, and Khai Truong. Automating clinical documentation with digital scribes: Understanding the impact on physicians. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, pages 1–12, 2021.

[152] Xiaoya Li, Jingrong Feng, Yuxian Meng, Qinghong Han, Fei Wu, and Jiwei Li. A unified mrc framework for named entity recognition. *arXiv preprint arXiv:1910.11476*, 2019.

[153] Xiaoya Li, Fan Yin, Zijun Sun, Xiayu Li, Arianna Yuan, Duo Chai, Mingxin Zhou, and Jiwei Li. Entity-relation extraction as multi-turn question answering. *arXiv preprint arXiv:1905.05529*, 2019.

[154] Elisa G. Liberati, Francesca Ruggiero, Laura Galuppo, Mara Gorli, Marien González-Lorenzo, Marco Maraldi, Pietro Ruggieri, Hernan Polo Friz, Giuseppe Scaratti, Koren H. Kwag, Roberto Vespignani, and Lorenzo Moja. What hinders

the uptake of computerized decision support systems in hospitals? A qualitative study and framework for implementation. *Implementation Science*, 2017. ISSN 17485908. doi: 10.1186/s13012-017-0644-2.

[155] Albee Y Ling, Allison W Kurian, Jennifer L Caswell-Jin, George W Sledge, Nigam H Shah, and Suzanne R Tamang. Using natural language processing to construct a metastatic breast cancer cohort from linked cancer registry and electronic medical records data. *JAMIA Open*, 2019. ISSN 2574-2531. doi: 10.1093/jamiaopen/ooz040.

[156] Todd Lingren, Louise Deleger, Katalin Molnar, Haijun Zhai, Jareen Meinzen-Derr, Megan Kaiser, Laura Stoutenborough, Qi Li, and Imre Solti. Evaluating the impact of pre-annotation on annotation speed and potential bias: Natural language processing gold standard development for clinical named entity recognition in clinical trial announcements. *Journal of the American Medical Informatics Association*, 2014. ISSN 1527974X. doi: 10.1136/amiajnl-2013-001837.

[157] Andy T Liu, Wei Xiao, Henghui Zhu, Dejiao Zhang, Shang-Wen Li, and Andrew Arnold. Qaner: Prompting question answering models for few-shot named entity recognition. *arXiv preprint arXiv:2203.01543*, 2022.

[158] Bingbin Liu, Pradeep Ravikumar, and Andrej Risteski. Contrastive learning of strong-mixing continuous-time stochastic processes. In *International Conference on Artificial Intelligence and Statistics*, pages 3151–3159. PMLR, 2021.

[159] Haokun Liu, Derek Tam, Mohammed Muqeeth, Jay Mohta, Tenghao Huang, Mohit Bansal, and Colin Raffel. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. *arXiv preprint arXiv:2205.05638*, 2022.

[160] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *arXiv preprint arXiv:2107.13586*, 2021.

[161] Peter J. Liu. Learning to write notes in electronic health records. *CoRR*, abs/1808.02622, 2018. URL `http://arxiv.org/abs/1808.02622`.

[162] Xuan Liu, Xiaoguang Wang, and Stan Matwin. Improving the interpretability of deep neural networks with knowledge distillation. *CoRR*, abs/1812.10924, 2018. URL `http://arxiv.org/abs/1812.10924`.

[163] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2018.

[164] Yen Fu Luo, Weiyi Sun, and Anna Rumshisky. MCN: A comprehensive corpus for medical concept normalization, 2019. ISSN 15320464.

[165] David Madras, Toniann Pitassi, and Richard Zemel. Predict responsibly: improving fairness and accuracy by learning to defer. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 6150–6160, 2018.

[166] Anil N. Makam, Holly J. Lanham, Kim Batchelor, Lipika Samal, Brett Moran, Temple Howell-Stampley, Lynne Kirk, Manjula Cherukuri, Noel Santini, Luci K. Leykum, and Ethan A. Halm. Use and satisfaction with key functions of a common commercial electronic health record: A survey of primary care providers. *BMC Medical Informatics and Decision Making*, 13(1):86, August 2013. ISSN 1472-6947. doi: 10.1186/1472-6947-13-86.

[167] Lena Mamykina, David K. Vawdrey, Peter D. Stetson, Kai Zheng, and George Hripcsak. Clinical documentation: Composition or synthesis? *Journal of the American Medical Informatics Association*, 19(6):1025–1031, November 2012. ISSN 1067-5027. doi: 10.1136/amiajnl-2012-000901.

[168] Matthew McDermott, Bret Nestor, Evan Kim, Wancong Zhang, Anna Goldenberg, Peter Szolovits, and Marzyeh Ghassemi. A comprehensive ehr timeseries pre-training benchmark. In *Proceedings of the Conference on Health, Inference, and Learning*, CHIL '21, page 257–278, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450383592. doi: 10.1145/3450439.3451877. URL `https://doi.org/10.1145/3450439.3451877`.

[169] Clement J. McDonald. The Barriers to Electronic Medical Record Systems and How to Overcome Them. *Journal of the American Medical Informatics Association*, 4(3):213–221, 1997. ISSN 1067-5027.

[170] Nir Menachemi and Taleah Collum. Benefits and drawbacks of electronic health record systems. *Risk management and healthcare policy*, 4:47–55, 05 2011. doi: 10.2147/RMHP.S12985.

[171] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26, 2013.

[172] Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. Rethinking the role of demonstrations: What makes in-context learning work? *arXiv preprint arXiv:2202.12837*, 2022.

[173] Swaroop Mishra, Daniel Khashabi, Chitta Baral, Yejin Choi, and Hannaneh Hajishirzi. Reframing instructional prompts to gptk's language. *arXiv preprint arXiv:2109.07830*, 2021.

[174] Ishan Misra, C Lawrence Zitnick, and Martial Hebert. Shuffle and learn: unsupervised learning using temporal order verification. In *European Conference on Computer Vision*, pages 527–544. Springer, 2016.

[175] Hossein Mobahi, Ronan Collobert, and Jason Weston. Deep learning from temporal coherence in video. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 737–744, 2009.

[176] Sungrim Moon, Serguei Pakhomov, Nathan Liu, James O Ryan, and Genevieve B Melton. A sense inventory for clinical abbreviations and acronyms created using clinical notes and medical dictionary resources. *Journal of the American Medical Informatics Association*, 21(2):299–307, 2014.

[177] Milad Moradi, Kathrin Blagec, Florian Haberl, and Matthias Samwald. Gpt-3 models are poor few-shot learners in the biomedical domain. *arXiv preprint arXiv:2109.02555*, 2021.

[178] Thomas Morton and Jeremy LaCivita. Wordfreak: An open tool for linguistic annotation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology: Demonstrations - Volume 4*, NAACL-Demonstrations '03, page 17–18, USA, 2003. Association for Computational Linguistics. doi: 10.3115/1073427.1073436. URL `https://doi.org/10.3115/1073427.1073436`.

[179] Danielle L Mowery, Brett R South, Lee Christensen, Jianwei Leng, Laura-Maria Peltonen, Sanna Salanterä, Hanna Suominen, David Martinez, Sumithra Velupillai, Noémie Elhadad, et al. Normalizing acronyms and abbreviations to aid patient understanding of clinical texts: Share/clef ehealth challenge 2013, task 2. *Journal of biomedical semantics*, 7(1):1–13, 2016.

[180] Amanda J Moy, Jessica M Schwartz, RuiJun Chen, Shirin Sadri, Eugene Lucas, Kenrick D Cato, and Sarah Collins Rossetti. Measurement of clinical documentation burden among physicians and nurses using electronic health records: A scoping review. *Journal of the American Medical Informatics Association*, (ocaa325), January 2021. ISSN 1527-974X. doi: 10.1093/jamia/ocaa325.

[181] Hussein Mozannar and David Sontag. Consistent estimators for learning to defer to an expert. In *International Conference on Machine Learning*, pages 7076–7087. PMLR, 2020.

[182] James Mullenbach, Jordan Swartz, T. Greg McKelvey, Hui Dai, and David Sontag. Knowledge Base Completion for Constructing Problem-Oriented Medical Records. *arXiv:2004.12905 [cs, stat]*, August 2020.

[183] Shawn N Murphy, Griffin Weber, Michael Mendis, Vivian Gainer, Henry C Chueh, Susanne Churchill, and Isaac Kohane. Serving the enterprise and beyond with informatics for integrating biology and the bedside (i2b2). *Journal of the American Medical Informatics Association*, 17(2):124–130, 2010.

[184] Luke Murray, Divya Gopinath, Monica Agrawal, Steven Horng, David Sontag, and David R Karger. Medknowts: Unified documentation and information

retrieval for electronic health records. In *The 34th Annual ACM Symposium on User Interface Software and Technology*, pages 1169–1183, 2021.

[185] Prakash M. Nadkarni, Lucila Ohno-Machado, and Wendy W. Chapman. Natural language processing: An introduction, 2011. ISSN 10675027.

[186] Nagarajan Natarajan, Inderjit S. Dhillon, Pradeep Ravikumar, and Ambuj Tewari. Learning with noisy labels. In *Advances in Neural Information Processing Systems*, 2013.

[187] Stuart J. Nelson, Kelly Zeng, John Kilbourne, Tammy Powell, and Robin Moore. Normalized names for clinical drugs: RxNorm at 6 years. *Journal of the American Medical Informatics Association*, 2011. ISSN 10675027. doi: 10.1136/amiajnl-2011-000116.

[188] Mark Neumann, Daniel King, Iz Beltagy, and Waleed Ammar. ScispaCy: Fast and Robust Models for Biomedical Natural Language Processing. 2019. doi: 10.18653/v1/w19-5034.

[189] Mark Neumann, Daniel King, Iz Beltagy, and Waleed Ammar. ScispaCy: Fast and Robust Models for Biomedical Natural Language Processing. In *Proceedings of the 18th BioNLP Workshop and Shared Task*, pages 319–327, Florence, Italy, August 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-5034. URL `https://www.aclweb.org/anthology/W19-5034`.

[190] Mark Neumann, Daniel King, Iz Beltagy, and Waleed Ammar. Scispacy: Fast and robust models for biomedical natural language processing. *CoRR*, abs/1902.07669, 2019. URL `http://arxiv.org/abs/1902.07669`.

[191] Aurélie Névéol, Hercules Dalianis, Sumithra Velupillai, Guergana Savova, and Pierre Zweigenbaum. Clinical natural language processing in languages other than english: opportunities and challenges. *Journal of biomedical semantics*, 9 (1):1–13, 2018.

[192] Andrew Y. Ng and Michael I. Jordan. *Shaping and Policy Search in Reinforcement Learning*. PhD thesis, 2003. AAI3105322.

[193] Jakob Nielsen. *Usability engineering*. Morgan Kaufmann an imprint of Academic Press, a Harcourt Science and Technology Company, 1993.

[194] NIH-NLM. SNOMED Clinical Terms® (SNOMED CT®), 2015.

[195] Mahsan Nourani, Joanie T. King, and Eric D. Ragan. The role of domain expertise in user trust and the impact of first impressions with intelligent systems, 2020.

[196] Benjamin Nye, Junyi Jessy Li, Roma Patel, Yinfei Yang, Iain J Marshall, Ani Nenkova, and Byron C Wallace. A corpus with multi-level annotations of patients, interventions and outcomes to support language processing for medical

literature. In *Proceedings of the conference. Association for Computational Linguistics. Meeting*, volume 2018, page 197. NIH Public Access, 2018.

[197] Philip V Ogren. Knowtator : A Protégé plug-in for annotated corpus construction. *Proceedings of the 2006 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology: companion volume: demonstrations*, 2006. doi: 10.3115/1225785.1225791.

[198] Jorge Piazentin Ono, Arvi Gjoka, Justin Salamon, Carlos Dietrich, and Claudio T. Silva. HistoryTracker: Minimizing human interactions in baseball game annotation. In *Conference on Human Factors in Computing Systems - Proceedings*, 2019. ISBN 9781450359702. doi: 10.1145/3290605.3300293.

[199] Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *arXiv preprint arXiv:2203.02155*, 2022.

[200] Anusri Pampari, Preethi Raghavan, Jennifer Liang, and Jian Peng. emrqa: A large corpus for question answering on electronic medical records. *arXiv preprint arXiv:1809.00732*, 2018.

[201] Andrea Papenmeier, Gwenn Englebienne, and Christin Seifert. How model accuracy and explanation fidelity influence user trust in AI, 2019.

[202] J. Parker and E. Coiera. Improving clinical communication: A view from psychology. *Journal of the American Medical Informatics Association: JAMIA*, 7(5):453–461, 2000 Sep-Oct. ISSN 1067-5027. doi: 10.1136/jamia.2000.0070453.

[203] Pinalkumar Patel, Disha Davey, Vishal Panchal, and Parth Pathak. Annotation of a large clinical entity corpus. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, EMNLP 2018*, 2018. ISBN 9781948087841. doi: 10.18653/v1/d18-1228.

[204] V. L. Patel, A. W. Kushniruk, S. Yang, and J. F. Yale. Impact of a computer-based patient record system on data collection, knowledge organization, and reasoning. *Journal of the American Medical Informatics Association: JAMIA*, 7 (6):569–585, 2000 Nov-Dec. ISSN 1067-5027. doi: 10.1136/jamia.2000.0070569.

[205] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[206] Neil Perry, Megha Srivastava, Deepak Kumar, and Dan Boneh. Do users write more insecure code with ai assistants? *arXiv preprint arXiv:2211.03622*, 2022.

[207] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1202. URL `https://aclanthology.org/N18-1202`.

[208] Brian W. Pickering, Yue Dong, Adil Ahmed, Jyothsna Giri, Oguz Kilickaya, Ashish Gupta, Ognjen Gajic, and Vitaly Herasevich. The implementation of clinician designed, human-centered electronic medical record viewer in the intensive care unit: A pilot step-wedge cluster randomized trial. *International Journal of Medical Informatics*, 84(5):299–307, May 2015. ISSN 13865056. doi: 10.1016/j.ijmedinf.2015.01.017.

[209] Emma Pierson, Pang Wei Koh, Tatsunori Hashimoto, Daphne Koller, Jure Leskovec, Nick Eriksson, and Percy Liang. Inferring multidimensional rates of aging from cross-sectional data. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 97–107. PMLR, 2019.

[210] Peter Pirolli and Stuart Card. Information foraging in information access environments. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '95, pages 51–58, USA, May 1995. ACM Press/Addison-Wesley Publishing Co. ISBN 978-0-201-84705-5. doi: 10.1145/223904.223911.

[211] Andrey A. Povyakalo, Eugenio Alberdi, Lorenzo Strigini, and Peter Ayton. How to discriminate between computer-aided and computer-hindered decisions: A case study in mammography. *Medical Decision Making*, 2013. ISSN 0272989X. doi: 10.1177/0272989X12465490.

[212] Sameer Pradhan, Noémie Elhadad, Wendy Chapman, Suresh Manandhar, and Guergana Savova. SemEval-2014 Task 7: Analysis of Clinical Text. 2015. doi: 10.3115/v1/s14-2007.

[213] Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D Manning. Stanza: A python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 101–108, 2020.

[214] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020. URL `http://jmlr.org/papers/v21/20-074.html`.

[215] Maithra Raghu, Katy Blumer, Greg Corrado, Jon Kleinberg, Ziad Obermeyer, and Sendhil Mullainathan. The algorithmic automation problem: Prediction, triage, and human effort. *arXiv preprint arXiv:1903.12220*, 2019.

[216] Jonathan Raiman and Olivier Raiman. DeepType: Multilingual entity linking by neural type system evolution. In *32nd AAAI Conference on Artificial Intelligence, AAAI 2018*, 2018. ISBN 9781577358008.

[217] Alvin Rajkomar, Eyal Oren, Kai Chen, Andrew M. Dai, Nissan Hajaj, Michaela Hardt, Peter J. Liu, Xiaobing Liu, Jake Marcus, Mimi Sun, Patrik Sundberg, Hector Yee, Kun Zhang, Yi Zhang, Gerardo Flores, Gavin E. Duggan, Jamie Irvine, Quoc Le, Kurt Litsch, Alexander Mossin, Justin Tansuwan, De Wang, James Wexler, Jimbo Wilson, Dana Ludwig, Samuel L. Volchenboum, Katherine Chou, Michael Pearson, Srinivasan Madabushi, Nigam H. Shah, Atul J. Butte, Michael D. Howell, Claire Cui, Greg S. Corrado, and Jeffrey Dean. Scalable and accurate deep learning with electronic health records. *arXiv*, (January): 1–10, 2018. ISSN 2398-6352. doi: 10.1038/s41746-018-0029-1. URL `http://dx.doi.org/10.1038/s41746-018-0029-1`.

[218] Alexander Ratner, Stephen H Bach, Henry Ehrenberg, Jason Fries, Sen Wu, and Christopher Ré. Snorkel: Rapid training data creation with weak supervision. In *Proceedings of the VLDB Endowment. International Conference on Very Large Data Bases*, volume 11, page 269. NIH Public Access, 2017.

[219] Ruth Reátegui and Sylvie Ratté. Comparison of MetaMap and cTAKES for entity extraction in clinical notes. *BMC Medical Informatics and Decision Making*, 2018. ISSN 14726947. doi: 10.1186/s12911-018-0654-2.

[220] A. L. Rector, W. A. Nowlan, and S. Kay. Foundations for an electronic medical record. *Methods of Information in Medicine*, 30(3):179–186, August 1991. ISSN 0026-1270.

[221] Alejandro Riveros, Maria De Arteaga, Fabio González, Sergio Jimenez, and Henning Müller. MindLab-UNAL: Comparing Metamap and T-mapper for Medical Concept Extraction in SemEval 2014 Task 7. 2015. doi: 10.3115/v1/s14-2073.

[222] Kirk Roberts. Assessing the corpus size vs. similarity trade-off for word embeddings in clinical nlp. In *Proceedings of the Clinical Natural Language Processing Workshop (ClinicalNLP)*, pages 54–63, 2016.

[223] Joshua Robinson, Ching-Yao Chuang, Suvrit Sra, and Stefanie Jegelka. Contrastive learning with hard negative samples. *arXiv preprint arXiv:2010.04592*, 2020.

[224] Joshua Robinson, Ching-Yao Chuang, Suvrit Sra, and Stefanie Jegelka. Contrastive learning with hard negative samples. *ICLR*, 2021.

[225] Ruth Reátegui Rojas and Sylvie Ratté. Comparison of metamap and ctakes for entity extraction in clinical notes. *BMC Medical Informatics and Decision Making*, 18, 2018.

[226] S Trent Rosenbloom, Joshua C Denny, Hua Xu, Nancy Lorenzi, William W Stead, and Kevin B Johnson. Data from clinical notes: A perspective on the tension between structure and flexible documentation. *Journal of the American Medical Informatics Association*, 18(2):181–186, March 2011. ISSN 1067-5027. doi: 10.1136/jamia.2010.007237.

[227] Peter M. Rothwell. External validity of randomised controlled trials: "to whom do the results of this trial apply?", 2005. ISSN 01406736.

[228] Maya Rotmensch, Yoni Halpern, Abdulhakim Tlimat, Steven Horng, and David Sontag. Learning a Health Knowledge Graph from Electronic Medical Records. *Scientific Reports*, 2017. ISSN 20452322. doi: 10.1038/s41598-017-05778-z.

[229] Adam Rule, Steven Rick, Michael Chiu, Phillip Rios, Shazia Ashfaq, Alan Calvitti, Wesley Chan, Nadir Weibel, and Zia Agha. Validating free-text order entry for a note-centric EHR. *AMIA Annual Symposium Proceedings*, 2015: 1103–1110, November 2015. ISSN 1942-597X.

[230] Adam Rule, Isaac H. Goldstein, Michael F. Chiang, and Michelle R. Hribar. Clinical Documentation as End-User Programming. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI '20, pages 1–13, New York, NY, USA, April 2020. Association for Computing Machinery. ISBN 978-1-4503-6708-0. doi: 10.1145/3313831.3376205.

[231] David L Sackett. Evidence-based medicine. In *Seminars in perinatology*, volume 21, pages 3–5. Elsevier, 1997.

[232] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *ArXiv*, abs/1910.01108, 2019.

[233] Victor Sanh, Albert Webson, Colin Raffel, Stephen H Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, et al. Multitask prompted training enables zero-shot task generalization. *arXiv preprint arXiv:2110.08207*, 2021.

[234] Suchi Saria, Atul Butte, and Aziz Sheikh. Better medicine through machine learning: What's real, and what's artificial?, 2018. ISSN 15491676.

[235] Nikunj Saunshi, Orestis Plevrakis, Sanjeev Arora, Mikhail Khodak, and Hrishikesh Khandeparkar. A theoretical analysis of contrastive unsupervised representation learning. In *International Conference on Machine Learning*, pages 5628–5637. PMLR, 2019.

[236] Guergana K. Savova, James J. Masanz, Philip V. Ogren, Jiaping Zheng, Sunghwan Sohn, Karin C. Kipper-Schuler, and Christopher G. Chute. Mayo clinical Text Analysis and Knowledge Extraction System (cTAKES): Architecture, component evaluation and applications. *Journal of the American Medical Informatics Association*, 2010. ISSN 10675027. doi: 10.1136/jamia.2009.001560.

[237] Guergana K Savova, James J Masanz, Philip V Ogren, Jiaping Zheng, Sunghwan Sohn, Karin C Kipper-Schuler, and Christopher G Chute. Mayo clinical Text Analysis and Knowledge Extraction System (cTAKES): Architecture, component evaluation and applications. *Journal of the American Medical Informatics Association*, 17(5):507–513, September 2010. ISSN 1067-5027. doi: 10.1136/jamia.2009.001560.

[238] Guergana K. Savova, James J. Masanz, Philip V. Ogren, Jiaping Zheng, Sunghwan Sohn, Karin Kipper Schuler, and Christopher G. Chute. Mayo clinical text analysis and knowledge extraction system (ctakes): architecture, component evaluation and applications. *Journal of the American Medical Informatics Association : JAMIA*, 17 5:507–13, 2010.

[239] Timo Schick and Hinrich Schütze. Exploiting cloze-questions for few-shot text classification and natural language inference. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 255–269, 2021.

[240] Benjamin Schloss and Sandeep Konam. Towards an automated soap note: classifying utterances from medical conversations. In *Machine Learning for Healthcare Conference*, pages 610–631. PMLR, 2020.

[241] Jeffrey L. Schnipper, Jeffrey A. Linder, Matvey B. Palchuk, Jonathan S. Einbinder, Qi Li, Anatoly Postilnik, and Blackford Middleton. "Smart Forms" in an Electronic Medical Record: Documentation-based Clinical Decision Support to Improve Disease Management. *Journal of the American Medical Informatics Association*, 15(4):513–523, July 2008. ISSN 1067-5027. doi: 10.1197/jamia.M2501.

[242] Fred Schulte and Erika Fry. Death by 1,000 clicks: Where electronic health records went wrong. *Kaiser Health News*, Jun 2019. URL `https://khn.org/news/death-by-a-thousand-clicks/`.

[243] D. Sculley, Gary Holt, Daniel Golovin, Eugene Davydov, Todd Phillips, Dietmar Ebner, Vinay Chaudhary, Michael Young, Jean François Crespo, and Dan Dennison. Hidden technical debt in machine learning systems. In *Advances in Neural Information Processing Systems*, 2015.

[244] Michael G Semanik, Peter C Kleinschmidt, Adam Wright, Duwayne L Willett, Shannon M Dean, Sameh N Saleh, Zoe Co, Emmanuel Sampene, and Joel R Buchanan. Impact of a problem-oriented view on clinical data retrieval. *Journal of the American Medical Informatics Association*, 28(5):899–906, May 2021. ISSN 1527-974X. doi: 10.1093/jamia/ocaa332.

[245] Burr Settles and Mark Craven. An analysis of active learning strategies for sequence labeling tasks. In *EMNLP 2008 - 2008 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference: A Meeting of SIGDAT, a Special Interest Group of the ACL*, 2008. doi: 10.3115/1613715.1613855.

[246] Rachel E. Sherman, Steven A. Anderson, Gerald J. Dal Pan, Gerry W. Gray, Thomas Gross, Nina L. Hunter, Lisa LaVange, Danica Marinac-Dabic, Peter W. Marks, Melissa A. Robb, Jeffrey Shuren, Robert Temple, Janet Woodcock, Lilly Q. Yue, and Robert M. Califf. Real-world evidence - What is it and what can it tell us? *New England Journal of Medicine*, 2016. ISSN 15334406. doi: 10.1056/NEJMsb1609216.

[247] Peng Shi and Jimmy Lin. Simple bert models for relation extraction and semantic role labeling. *arXiv preprint arXiv:1904.05255*, 2019.

[248] Lotan Shilo and Gila Shilo. Analysis of abbreviations used by residents in admission notes and discharge summaries. *QJM: An International Journal of Medicine*, 111(3):179–183, 2018.

[249] Yuqi Si, Jingqi Wang, Hua Xu, and Kirk Roberts. Enhancing clinical concept extraction with contextual embeddings. *Journal of the American Medical Informatics Association*, 2019. ISSN 1527974X. doi: 10.1093/jamia/ocz096.

[250] Pao Siangliulue, Joel Chan, Krzysztof Z. Gajos, and Steven P. Dow. Providing timely examples improves the quantity and quality of generated ideas. In *Proceedings of the 2015 ACM SIGCHI Conference on Creativity and Cognition*, page 83–92, New York, NY, USA, 2015. Association for Computing Machinery. ISBN 9781450335980. doi: 10.1145/2757226.2757230. URL `https://doi.org/10.1145/2757226.2757230`.

[251] James E. Siegler, Neha N. Patel, and C. Jessica Dine. Prioritizing Paperwork Over Patient Care: Why Can't We Do Both? *Journal of Graduate Medical Education*, 7(1):16–18, March 2015. ISSN 1949-8349. doi: 10.4300/JGME-D-14-00494.1.

[252] Pulkit Singh, Julian Haimovich, Christopher Reeder, Shaan Khurshid, Emily S Lau, Jonathan W Cunningham, Anthony Philippakis, Christopher D Anderson, Jennifer E Ho, Steven A Lubitz, et al. One clinician is all you need–cardiac magnetic resonance imaging measurement extraction: Deep learning algorithm development. *JMIR Medical Informatics*, 10(9):e38178, 2022.

[253] Karan Singhal, Shekoofeh Azizi, Tao Tu, S Sara Mahdavi, Jason Wei, Hyung Won Chung, Nathan Scales, Ajay Tanwani, Heather Cole-Lewis, Stephen Pfohl, et al. Large language models encode clinical knowledge. *arXiv preprint arXiv:2212.13138*, 2022.

[254] C. Smith, S. Hetzel, P. Dalrymple, and A. Keselman. Beyond readability: Investigating coherence of clinical text for consumers. *Journal of medical Internet research*, 13:e104, 10 2011. doi: 10.2196/jmir.1842.

[255] C. A. Smith, S. Hetzel, P. Dalrymple, and A. Keselman. Beyond readability: investigating coherence of clinical text for consumers. *Journal of Medical Internet Research*, 13(4):e104, December 2011.

[256] Ryan Smith, Jason A Fries, Braden Hancock, and Stephen H Bach. Language models in the loop: Incorporating prompting into weak supervision. *arXiv preprint arXiv:2205.02318*, 2022.

[257] Sunghwan Sohn, Cheryl Clark, Scott R Halgrim, Sean P Murphy, Christopher G Chute, and Hongfang Liu. Medxn: an open source medication extraction and normalization tool for clinical text. *Journal of the American Medical Informatics Association*, 21(5):858–865, 2014.

[258] Brett R South, Danielle Mowery, Ying Suo, Jianwei Leng, Oscar Ferrandez, Stephane M Meystre, and Wendy Champan. Evaluating the effects of machine pre-annotation and an interactive annotation interface on manual deidentification of clinical text, 2014.

[259] Ethan Steinberg, Ken Jung, Jason A Fries, Conor K Corbin, Stephen R Pfohl, and Nigam H Shah. Language models are an effective representation learning technique for electronic health record data. *Journal of Biomedical Informatics*, 113:103637, 2021.

[260] Pontus Stenetorp, Sampo Pyysalo, Goran Topíc, Tomoko Ohta, Sophia Ananiadou, and Jun'ichi Tsujii. BRAT: AWeb-based tool for NLP-Assisted text annotation. In *EACL 2012 - Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics*, 2012. ISBN 9781937284190.

[261] Amber Stubbs. Mae and mai: lightweight annotation and adjudication tools. In *Proceedings of the 5th linguistic annotation workshop*, pages 129–133, 2011.

[262] Nicole Sultanum, Devin Singh, Michael Brudno, and Fanny Chevalier. *Doccurate* : A Curation-Based Approach for Clinical Text Visualization. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):142–151, January 2019. ISSN 1077-2626, 1941-0506, 2160-9306. doi: 10.1109/TVCG.2018.2864905.

[263] Weiyi Sun, Anna Rumshisky, and Ozlem Uzuner. Evaluating temporal relations in clinical text: 2012 i2b2 challenge. *Journal of the American Medical Informatics Association*, 20(5):806–813, 2013.

[264] Hanna Suominen, Sanna Salanterä, Sumithra Velupillai, Wendy W. Chapman, Guergana Savova, Noemie Elhadad, Sameer Pradhan, Brett R. South, Danielle L. Mowery, Gareth J.F. Jones, Johannes Leveling, Liadh Kelly, Lorraine Goeuriot, David Martinez, and Guido Zuccon. Overview of the ShARe/CLEF eHealth evaluation lab 2013. *Lecture Notes in Computer Science*, 8138 LNCS:212–231, 2013. ISSN 03029743. doi: 10.1007/978-3-642-40802-1_24.

[265] Harini Suresh, Natalie Lao, and Ilaria Liccardi. Misplaced Trust: Measuring the Interference of Machine Learning in Human Decision-Making. In *WebSci 2020 - Proceedings of the 12th ACM Conference on Web Science*, 2020. ISBN 9781450379892. doi: 10.1145/3394231.3397922.

[266] H. J. Tange, A. Hasman, P. F. de Vries Robbe, and H. C. Schouten. Medical narratives in electronic medical records. *International Journal of Medical Informatics*, 46(1):7–29, Aug 1997.

[267] H. J. Tange, A. Hasman, P. F. de Vries Robbé, and H. C. Schouten. Medical narratives in electronic medical records. *International Journal of Medical Informatics*, 46(1):7–29, August 1997. ISSN 1386-5056. doi: 10.1016/s1386-5056(97)00048-8.

[268] Huibert Tange. How to approach the structuring of the medical record? Towards a model for flexible access to free text medical data. *International Journal of Bio-Medical Computing*, 42(1):27–34, July 1996. ISSN 0020-7101. doi: 10.1016/0020-7101(96)01178-6.

[269] Derjung M Tarn and Susan A Flocke. New prescriptions: how well do patients remember important information? *Family medicine*, 43(4):254, 2011.

[270] Katrin Tomanek and Udo Hahn. Semi-supervised active learning for sequence labeling. 2009. doi: 10.3115/1690219.1690291.

[271] Christopher Tosh, Akshay Krishnamurthy, and Daniel Hsu. Contrastive estimation reveals topic posterior information to linear models. *Journal of Machine Learning Research*, 22(281):1–31, 2021.

[272] Christopher Tosh, Akshay Krishnamurthy, and Daniel Hsu. Contrastive learning, multi-view redundancy, and linear models. In *Algorithmic Learning Theory*, pages 1179–1206. PMLR, 2021.

[273] Shubham Toshniwal, Sam Wiseman, Allyson Ettinger, Karen Livescu, and Kevin Gimpel. Learning to ignore: Long document coreference with bounded memory neural networks. *arXiv preprint arXiv:2010.02807*, 2020.

[274] Shubham Toshniwal, Patrick Xia, Sam Wiseman, Karen Livescu, and Kevin Gimpel. On generalization in coreference resolution. *arXiv preprint arXiv:2109.09667*, 2021.

[275] Michael A. Tutty, Lindsey E. Carlasare, Stacy Lloyd, and Christine A. Sinsky. The complex case of EHRs: Examining the factors impacting the EHR user experience. *Journal of the American Medical Informatics Association: JAMIA*, 26(7):673–677, July 2019. ISSN 1527-974X. doi: 10.1093/jamia/ocz021.

[276] Özlem Uzuner, Imre Solti, and Eithon Cadag. Extracting medication information from clinical text. *Journal of the American Medical Informatics Association*, 17 (5):514–518, 2010.

[277] Özlem Uzuner, Brett R. South, Shuying Shen, and Scott L. DuVall. 2010 i2b2/VA challenge on concepts, assertions, and relations in clinical text. *Journal of the American Medical Informatics Association*, 2011. ISSN 10675027. doi: 10.1136/amiajnl-2011-000203.

[278] Ozlem Uzuner, Andreea Bodnari, Shuying Shen, Tyler Forbush, John Pestian, and Brett R South. Evaluating the state of the art in coreference resolution for electronic medical records. *Journal of the American Medical Informatics Association*, 19(5):786–791, 2012.

[279] A. M. van Ginneken. The Physician's Flexible Narrative. *Methods of Information in Medicine*, 35(2):98–100, 1996. ISSN 0026-1270, 2511-705X. doi: 10.1055/s-0038-1634651.

[280] T. T. Van Vleck, A. Wilcox, P. D. Stetson, S. B. Johnson, and N. Elhadad. Content and structure of clinical problem lists: a corpus analysis. *AMIA Annual Symposium Proceedings*, pages 753–757, Nov 2008.

[281] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017.

[282] Mathias Verbeke, Vincent Van Asch, Roser Morante, Paolo Frasconi, Walter Daelemans, and Luc De Raedt. A statistical relational learning approach to identifying evidence based medicine categories. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 579–589, 2012.

[283] C. Jason Wang. Medical documentation in the electronic era, 2012. ISSN 00987484.

[284] Dayong Wang, Aditya Khosla, Rishab Gargeya, Humayun Irshad, and Andrew H Beck. Deep learning for identifying metastatic breast cancer. *arXiv preprint arXiv:1606.05718*, 2016.

[285] Liwei Wang, Jason Wampfler, Angela Dispenzieri, Hua Xu, and Hongfang Liu. Achievability to Extract Specific Date Information for Cancer Research. *Proceedings of AMIA Annual Symposium*, pages 893–902, 2019.

[286] Shuohang Wang, Yang Liu, Yichong Xu, Chenguang Zhu, and Michael Zeng. Want to reduce labeling cost? gpt-3 can help. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4195–4205, 2021.

[287] Yanshan Wang, Liwei Wang, Majid Rastegar-Mojarad, Sungrim Moon, Feichen Shen, Naveed Afzal, Sijia Liu, Yuqun Zeng, Saeed Mehrabi, Sunghwan Sohn, and Hongfang Liu. Clinical information extraction applications: A literature review, 2018. ISSN 15320464.

[288] Yanshan Wang, Liwei Wang, Majid Rastegar-Mojarad, Sungrim Moon, Feichen Shen, Naveed Afzal, Sijia Liu, Yuqun Zeng, Saeed Mehrabi, Sunghwan Sohn, et al. Clinical information extraction applications: a literature review. *Journal of biomedical informatics*, 77:34–49, 2018.

[289] Marc Weeber, James G Mork, and Alan R Aronson. Developing a test collection for biomedical word sense disambiguation. In *Proceedings of the AMIA Symposium*, page 746. American Medical Informatics Association, 2001.

[290] L. L. Weed. Medical records that guide and teach. *The New England Journal of Medicine*, 278(11):593–600, March 1968. ISSN 0028-4793. doi: 10.1056/NEJM196803142781105.

[291] Donglai Wei, Joseph J Lim, Andrew Zisserman, and William T Freeman. Learning and using the arrow of time. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8052–8060, 2018.

[292] Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. Finetuned language models are zero-shot learners. *arXiv:2109.01652*, 2021.

[293] Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682*, 2022.

[294] J.M. Weis and P.C. Levy. Copy, paste, and cloned notes in electronic health records. *Chest*, 145(3):632–638, 2014.

[295] Justin M. Weis and Paul C. Levy. Copy, Paste, and Cloned Notes in Electronic Health Records. *CHEST*, 145(3):632–638, March 2014. ISSN 0012-3692. doi: 10.1378/chest.13-0886.

[296] Wei-Hung Weng, Yu-An Chung, and Peter Szolovits. Unsupervised clinical language translation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 3121–3131, 2019.

[297] Adam B. Wilcox, Scott P. Narus, and Watson A. Bowes. Using natural language processing to analyze physician modifications to data entry templates. *Proceedings of the AMIA Symposium*, pages 899–903, 2002. ISSN 1531-605X.

[298] Adam B. Wilcox, Yueh-Hsia Chen, and George Hripcsak. Minimizing electronic health record patient-note mismatches. *Journal of the American Medical Informatics Association: JAMIA*, 18(4):511–514, 2011 Jul-Aug. ISSN 1527-974X. doi: 10.1136/amiajnl-2010-000068.

[299] Lauren Wilcox, Jie Lu, Jennifer Lai, Steven Feiner, and Desmond Jordan. Physician-driven management of patient progress notes in an intensive care unit. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1879–1888. Association for Computing Machinery, New York, NY, USA, April 2010. ISBN 978-1-60558-929-9.

[300] Bryan Wilder, Eric Horvitz, and Ece Kamar. Learning to complement humans. *arXiv preprint arXiv:2005.00582*, 2020.

[301] Ronald J. Willia. Simple Statistical Gradient-Following Algorithms for Connectionist Reinforcement Learning. *Machine Learning*, 1992. ISSN 15730565. doi: 10.1023/A:1022672621406.

[302] Laurenz Wiskott and Terrence J Sejnowski. Slow feature analysis: Unsupervised learning of invariances. *Neural computation*, 14(4):715–770, 2002.

[303] Jacob O. Wobbrock, Leah Findlater, Darren Gergle, and James J. Higgins. The Aligned Rank Transform for nonparametric factorial analyses using only ANOVA procedures. In *Conference on Human Factors in Computing Systems - Proceedings*, 2011. ISBN 9781450302289. doi: 10.1145/1978942.1978963.

[304] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*, 2019.

[305] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online, October 2020. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/2020.emnlp-demos.6`.

[306] S. Wu, K. Roberts, S. Datta, J. Du, Z. Ji, Y. Si, S. Soni, Q. Wang, Q. Wei, Y. Xiang, B. Zhao, and H. Xu. Deep learning in clinical natural language processing: a methodical review. *Journal of American Medical Informatics Association*, 27(3):457–470, Mar 2020.

[307] Stephen Wu, Kirk Roberts, Surabhi Datta, Jingcheng Du, Zongcheng Ji, Yuqi Si, Sarvesh Soni, Qiong Wang, Qiang Wei, Yang Xiang, et al. Deep learning in clinical natural language processing: a methodical review. *Journal of the American Medical Informatics Association*, 27(3):457–470, 2020.

[308] Tongshuang Wu, Michael Terry, and Carrie Jun Cai. Ai chains: Transparent and controllable human-ai interaction by chaining large language model prompts. In *CHI Conference on Human Factors in Computing Systems*, pages 1–22, 2022.

[309] Yonghui Wu, Joshua C. Denny, S. Trent Rosenbloom, Randolph A. Miller, Dario A. Giuse, and Hua Xu. A comparative study of current Clinical Natural Language Processing systems on handling abbreviations in discharge summaries. *AMIA ... Annual Symposium proceedings / AMIA Symposium. AMIA Symposium*, 2012. ISSN 1942597X.

[310] Yonghui Wu, Jun Xu, Yaoyun Zhang, and Hua Xu. Clinical abbreviation disambiguation using neural word embeddings. In *Proceedings of BioNLP 15*, pages 171–176, 2015.

[311] Fei Xia and Meliha Yetisgen-Yildiz. Clinical corpus annotation: challenges and strategies. In *Proceedings of the Third Workshop on Building and Evaluating Resources for Biomedical Text Mining (BioTxtM'2012) in conjunction with the International Conference on Language Resources and Evaluation (LREC), Istanbul, Turkey*, 2012.

[312] Jie Yang, Yue Zhang, Linwei Li, and Xingxuan Li. YEDDA: A lightweight collaborative text span annotation tool. In *ACL 2018 - 56th Annual Meeting of the Association for Computational Linguistics, Proceedings of System Demonstrations*, 2018. ISBN 9781948087650. doi: 10.18653/v1/p18-4006.

[313] Qian Yang, Aaron Steinfeld, and John Zimmerman. Unremarkable AI: Fitting Intelligent Decision Support into Critical, Clinical Decision-Making Processes. *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pages 1–11, May 2019. doi: 10.1145/3290605.3300468.

[314] Xiaohan Yang, Eduardo Peynetti, Vasco Meerman, and Chris Tanner. What gpt knows about who is who. *arXiv preprint arXiv:2205.07407*, 2022.

[315] Seid Muhie Yimam, Chris Biemann, Richard Eckart de Castilho, and Iryna Gurevych. Automatic annotation suggestions and custom annotation layers in WebAnno. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 91–96, Baltimore, Maryland, June 2014. Association for Computational Linguistics. doi: 10.3115/v1/P14-5016. URL `https://www.aclweb.org/anthology/P14-5016`.

[316] Ming Yin, Jennifer Wortman Vaughan, and Hanna Wallach. Understanding the effect of accuracy on trust in machine learning models. In *Conference on Human Factors in Computing Systems - Proceedings*, 2019. ISBN 9781450359702. doi: 10.1145/3290605.3300509.

[317] Jinsung Yoon, Yao Zhang, James Jordon, and Mihaela van der Schaar. Vime: Extending the success of self-and semi-supervised learning to tabular domain. *Advances in Neural Information Processing Systems*, 33, 2020.

[318] Xiang Yue, Xinliang Frederick Zhang, Ziyu Yao, Simon Lin, and Huan Sun. Cliniqg4qa: Generating diverse questions for domain adaptation of clinical question answering. In *2021 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 580–587. IEEE, 2021.

[319] Q. Zeng and J. J. Cimino. A knowledge-based, concept-oriented view generation system for clinical data. *Journal of Biomedical Informatics*, 34(2):112–128, April 2001. ISSN 1532-0464. doi: 10.1006/jbin.2001.1013.

[320] Jieyu Zhang, Yue Yu, Yinghao Li, Yujing Wang, Yaming Yang, Mao Yang, and Alexander Ratner. Wrench: A comprehensive benchmark for weak supervision. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021.

[321] Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*, 2022.

[322] Tianyi Zhang, Felix Wu, Arzoo Katiyar, Kilian Q. Weinberger, and Yoav Artzi. Revisiting few-sample BERT fine-tuning. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL `https://openreview.net/forum?id=cO1IH43yUF`.

[323] Yunfeng Zhang, Q. Vera Liao, and Rachel K.E. Bellamy. Efect of confidence and explanation on accuracy and trust calibration in AI-assisted decision making. In *FAT* 2020 - Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, 2020. ISBN 9781450369367. doi: 10.1145/3351095.3372852.

[324] Jason Zhao, Monica Agrawal, Pedram Razavi, and David Sontag. Directing human attention in event localization for clinical timeline creation. In *Machine Learning for Healthcare Conference*, pages 80–102. PMLR, 2021.

[325] Jiaping Zheng, Wendy W Chapman, Rebecca S Crowley, and Guergana K Savova. Coreference resolution: A review of general methodologies and applications in the clinical domain. *Journal of biomedical informatics*, 44(6):1113–1122, 2011.

[326] Kai Zheng, Rema Padman, Michael P. Johnson, and Herbert S. Diamond. An interface-driven analysis of user interactions with an electronic health records system. *Journal of the American Medical Informatics Association: JAMIA*, 16 (2):228–237, 2009 Mar-Apr. ISSN 1067-5027. doi: 10.1197/jamia.M2852.