Explainable query generation for cohort discovery and biomedical reasoning using natural language

Nicholas J. Dobbins

A dissertation
submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

University of Washington

2023

Reading Committee:
Meliha Yetisgen, Chair
H. Nina Kim, MD, MSc
Trevor Cohen, MBChB, PhD

Program Authorized to Offer Degree:
Biomedical and Health Informatics

University of Washington

Abstract

Explainable query generation for cohort discovery
and biomedical reasoning using natural language

Nicholas J. Dobbins

Chair of the Supervisory Committee:
Professor Meliha Yetisgen
Biomedical and Health Informatics

Clinical trials serve a critical role in the generation of medical evidence and enabling biomedical research. In order to identify potential participants, investigators publish eligibility criteria, such as past history of certain conditions, treatments, or laboratory tests. Patients meeting a trial's eligibility criteria are considered potential candidates for recruitment. Recruitment of participants remains, however, a major barrier to successful trial completion, and manual chart review of hundreds or thousands of patients to determine a candidate pool can be prohibitively labor- and time-intensive.

At the same time, the amount and variety of data contained in Electronic Health Records (EHRs) is increasing dramatically, creating both challenges and opportunities for patient recruitment. While more granular and potentially useful data are captured and stored in EHRs now than in the past, the process of accessing and leveraging these data often requires technical expertise and extensive knowledge of biomedical terminologies and data models.

This thesis focuses on the development of an integrated system for identifying patients in clinical databases using a natural language interface. Humans use natural language nearly effortlessly, and thus automated means of leveraging natural language to identify patients in databases hold great potential in time and cost savings. The primary contributions of this work include a novel database schema annotation and mapping method enabling data

model agnostic query generation, a method for generating intermediate logical representations of eligibility criteria, exploration of dynamic reasoning upon non-specific criteria, and development of an integrated graph-based knowledge base of biomedical concepts.

This work also introduces two new annotated corpora, the Leaf Clinical Trials (LCT) corpus and Leaf Logical Forms (LLF) corpus. The LCT corpus is unique in the granularity with which it represents complex eligibility criteria, while the LLF corpus is the most extensive annotated corpus of eligibility criteria logical representations at the time of this writing. Both corpora are valuable contributions to the biomedical informatics and natural language processing communities.

To evaluate the viability of our methods, both our system and a human database programmer generated queries to identify patients eligible for 8 past clinical trials at our institution. We then compared actual participant enrollments to those found eligible. We demonstrate that our system rivals and sometimes surpasses an experienced human programmer in finding eligible patients. We finally developed a novel user interface for enabling real-time interactive cohort discovery.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ALGORITHMS

LIST OF ABBREVIATIONS

AST: Abstract syntax tree

BI-LSTM: Bi-directional long-short term memory

CRF: Conditional random field

EHR: Electronic health record

FHIR: Fast healthcare interoperability resources

ICD: International classification of disease

KB: Knowledge base

LCT: Leaf clinical trials (corpus)

LLF: Leaf logical forms (corpus)

NLP: Natural language processing

OMOP: Observational medical outcomes partnership

SEQ2SEQ: Sequence to Sequence

SNOMED: Systematized nomenclature of medicine

SMM: Semantic metadata mapping

UMLS: Unified medical language system

ACKNOWLEDGMENTS

I want to begin by thanking my advisor, Meliha Yetisgen. At a time when I knew next to nothing about natural language processing, she kindly and patiently met with me and explained the basics and suggested what I should focus on learning. She was among the first to suggest I get a PhD, wrote not one but two recommendations for my graduate school application (first as non-matriculated, then as a formal PhD student), and has provided ongoing constructive feedback, advice, support, and opportunities for much of my career. I think there are few people in life that we can objectively say we have been changed by, but in mine she is certainly one of them. I'm eternally grateful.

I also want to thank the members of my committee, Nina Kim, Trevor Cohen, and Fei Xia. Nina was among the first physicians I interacted with at UW, and her feedback and suggestions over the years have been extremely helpful. Trevor and Fei are both truly excellent teachers and passionate about their work.

Sean Mooney and Peter Tarczy-Hornoch have also been critical to my PhD journey. I'll never forget the day that I showed Peter—who happened to be passing by—a demo of the prototype of Leaf. That Peter was willing to sit and talk with me about how it worked for over an hour meant the world to me (particularly as a former History major with never-ending imposter syndrome). Sean has been unfailingly supportive and a wonderful mentor.

Last, I'm incredibly grateful to my wife, Akiko. Through the years of working full-time while pursuing my PhD, she has helped me stay grounded and focused on our family and children, who I am intensely proud of and who will always be my greatest joy and priority. I can only hope that they grow to enjoy the same love of learning and imagining *what could be?* that I think make life interesting and worth living.

DEDICATION

To my children, Elly, Ami, and Sage.

Chapter 1
INTRODUCTION

1.1 Problem

Randomized controlled trials serve a critical role in the generation of medical evidence and furthering of biomedical research. In order to identify patients for clinical trials, investigators publish eligibility criteria, such as past history of certain conditions, treatments, or laboratory tests. These eligibility criteria are typically composed in free-text, consisting of inclusion and exclusion criteria. Patients meeting a trial's eligibility criteria are considered potential candidates for recruitment.

Recruitment of participants remains a major barrier to successful trial completion [1], so generating a large pool of potential candidates is often necessary. Manual chart review of hundreds or thousands of patients to determine a candidate pool, however, can be prohibitively labor- and time-intensive. Cohort discovery tools such as Leaf [2] and i2b2 [3] may be used, providing researchers with a relatively simple drag-and-drop graphical interface in their web browser to create database queries to find potential patients in electronic health records (EHR) [4]. Learning how to use such tools nevertheless presents a challenge, as graphically-represented concepts may not align with researchers' understanding of biomedical phenomena or trial eligibility criteria. In addition, certain complex queries may simply be impossible to execute due to structural limitations on the types of possible queries presented in these tools, such as complex temporal or nested sequences of events [5].

An alternative approach is the use of natural language processing (NLP) to automatically analyze eligibility criteria and generate database queries to find patients in EHRs. NLP-based approaches have the advantage of obviating potential learning curves of tools such as Leaf, while leveraging existing eligibility criteria composed in a format researchers are

already familiar with. Previous research in the use of NLP for trial recruitment [6, 7, 8, 9, 10, 11, 12, 13, 14] has shown promise but also limitations, preventing more widespread use for recruitment. For example, eligibility criteria often refer to non-specific criteria [15, 16], such as "conditions affecting respiratory function", which save investigator time by avoiding the need for exhaustive lists of conditions, but also must be reasoned upon to determine what conditions or concepts refer to. Past research has also largely supported only single, static data models, limiting potential usage to only institutions or researchers with data in particular formats, and preventing future additions or changes to data structure necessitated by changes in clinical workflow, research needs, and so on. In terms of evaluation, past research has also largely evaluated solutions based on metrics such as concept normalization performance, measuring how well applications can map text to coded elements such as ICD-10 or LOINC. While important, these metrics do not address the more important question of how well such tools are able to find patients meeting a given set of criteria.

This work explores the development of a user-friendly, explainable, and generalizable application for cohort discovery from natural language by querying virtually any database schema. We demonstrate that this system is capable of state-of-the-art query generation. We hope believe this system may enable faster discovery of eligible patients for clinical trials and biomedical research in general.

The importance of improving efficiencies in trial recruitment is difficult to overstate. Clinical trials are considered the gold standard of evaluation of new treatments but are also extremely costly; for example, the average cost of a Phase 3 trial in the United States ranges from US$11.5 million to US$52.9 million [17]. Much of those costs are due to difficulties in finding patients and keeping them enrolled. Systems which stand to improve efficiencies in finding patients eligible for trials using a medium researchers already know well and frequently use - natural language - therefore may also contribute to reducing costs and simplifying processes for enrolling patients.

The potential impact of this work extends beyond clinical trials. Biomedical research in general very often relies on finding patients meeting certain criteria, whether for retrospec-

tive analysis, sample size estimates, preparation to research, and so on. We hope that the methods and application we develop for creating a natural language interface to databases may simplify and streamline this common critical step for a variety of research purposes.

1.2 Contributions

This work explores the development of a novel database query-generation system to identify patients eligible for clinical trials using a natural language interface. The primary contributions of this work are:

1. *Annotated Corpora*: this work introduces two new annotated clinical corpora: the LCT corpus and LLF corpus. The LCT corpus contains over 1,000 annotated eligibility criteria documents and captures annotated phenomena important for the task of clinical query generation at a highly granular level [18]. These phenomena include clinical events, such as diagnoses, procedures, laboratory tests, vitals, and medications, demographic facets such as sex, ethnicity, and age, as well as abstract, temporal, and qualifying values such as contraindications, indications, temporal sequences, negations, risks, and severities. The LCT corpus is unique in the granularity of its annotation schema and number of annotated documents. The LLF corpus is a subset of 2,000 lines of eligibility criteria from the LCT corpus which includes the original criteria sentence alongside machine-readable intermediate logical representations, or logical forms. To the best of our knowledge, the LLF corpus is the only large publicly available data set of human-annotated clinical trial eligibility criteria as logical forms.

2. *Model-agnostic query generation*: We introduce a novel database annotation schema and mapping method to enable query generation on virtually any clinical data model. Our system uses a subset of Unified Medical Language System (UMLS) [19] concepts to flexibly represent database resources using concepts related to metadata, such as patient identifiers, demographics, and vocabularies.

3. *Knowledge Base*: Various tasks related to query generation, such as hierarchy navigation (e.g., finding types of surgeries related to cancers), vocabulary and code normalization (e.g., conversion to SNOMED, LOINC, or ICD-10 codes), and reasoning (explained next) necessitate the use of a Knowledge Base (KB). We introduce a graph-based KB which incorporates the UMLS, the Disease Ontology [20], Symptom Ontology [21], COVID-19 Ontology [22], Potential Drug-Drug Interactions [23], LOINC2HPO [24], and the Disease-Symptom Knowledge Base [25], as well as various equivalency mappings not present in the UMLS [26, 27, 28, 29].

4. *Reasoning upon non-specific criteria*: Many criteria within eligibility criteria are non-specific, and instead need to be first reasoned upon in order to be computable. For example, a criterion may refer to patients "indicated for bariatric surgery", or "who have a disease affecting respiratory function". Using our logical form representations and KB, we demonstrate a system for dynamically reasoning on non-specific criteria.

5. *Evaluation against a human programmer and actual trial enrollments*: Past systems for automatically identifying patients eligible for clinical trials have largely limited their evaluations to sub-systems such as normalization performance, and thus it is difficult to assess how well such systems actually perform when used with real-world data and trials. In contrast, we evaluate our system using enrollments linked to clinical data for 8 past clinical trials at the University of Washington. We compare the patients found by our generated queries to actual enrolled patients. We then compare our results to those of a human clinical database programmer.

6. *An interactive web application interface*: we enable interaction with our query generation system via an interactive web-based user interface. Our web application includes a novel chat-like interface system which incorporates streaming interfaces for rapid feedback. The interface enables users to view and edit system interpretations of natural language criteria in an iterative fashion.

1.3 Overview

Chapter 2, *Background*, provides a general overview of clinical trials, as well as past work in cohort discovery and eligibility prediction in relation to NLP. Literature specific to given tasks is also presented briefly in each chapter where applicable.

Chapter 3, *Leaf Clinical Trials corpus*, introduces the first annotated corpus, including motivation, annotation schema, comparison to other corpora, and baseline named entity recognition (NER) and relation extraction performance using multiple neural architectures.

Chapter 4, *Leaf Logical Forms corpus*, presents the second annotated corpus and our logical forms annotation schema. The corpus is evaluated using various syntax styles suggested from the literature.

Chapter 5, *Knowledge Base*, explores the motivation and development of the KB. We discuss each source at a high level and SPARQL queries used for reasoning.

Chapter 6, *Semantic Metadata Mapping*, describes the motivation and methods we use for dynamically generating queries for eligibility criteria in a data model-agnostic fashion.

Chapter 7, *Query Generation*, explores the development of our methods for query generation, including models used for NER and logical form transformation, normalization, and reasoning. Our system is evaluated in comparison to that of a human database programmer.

Chapter 8, *Web Application*, examines development of our user-facing web application for interactive cohort discovery.

Chapter 9, *Conclusions*, summarizes this work's primary contributions and possible directions for future research.

Chapter 2
BACKGROUND

This chapter provides an introduction to clinical trials and their importance to biomedical research. Section 2.1 provides background for and discusses the motivation and reasoning in focusing on clinical trials. Section 2.2 examines challenges in trial recruitment. Section 2.3 discusses past work in patient matching for clinical trials using electronic software. Section 2.4 examines challenges in electronic screening for trials using NLP. Section 2.5 discusses research in NLP methods for clinical trial recruitment, while Sections 2.6 and 2.7 examine published corpora related to clinical trials and clinical knowledge bases. Section 2.8 provides a summary of the chapter.

2.1 The Importance of Clinical Trials

A clinical trial is a prospective study comparing the effects and value of an intervention (typically a medication, biologic, or procedure) against a control group without it [30]. Clinical trials are considered "controlled" when a control group not receiving an interventional treatment is used for comparison, and "randomized" when participants are randomly placed in said treatment or control groups. Randomization is considered ideal for reducing risk of investigator bias and producing study groups closely in proportion to known risk factors. Randomized controlled trials (RCTs) are widely recognized as the best available method to determine if a given intervention is safe and effective [30].

Eligibility for a clinical trial is determined by a trial's *eligibility criteria*, which are free-text descriptions of required conditions, treatments, laboratory tests and so on. Eligibility criteria are composed of *inclusions*, which patients *must meet*, and *exclusions*, which patients *must not meet* in order to be eligible. The extent to which trial results can be assumed to

be generalizable to patients not participating in a trial but with comparable health status is determined by *external validity* [31]. External validity can be influenced by a number of factors, first and foremost the selection of patients potentially eligible for a trial. Finding patients appropriately meeting eligibility criteria in as unbiased way as possible is thus critical to drawing meaningful conclusions from clinical trials, and ultimately scientific progress and improvement of human health.

2.2 Challenges in Recruitment for Clinical Trials

Many clinical trials fail to meet their expected number of enrollments [32, 33, 34, 35, 36, 37]. The reasons for this are varied, including overly restrictive inclusion criteria [33], a lack of awareness on the part of patients, particularly in underserved and historically disadvantaged communities [35], fear or apprehension of medical research due to past abuses [32], and uncertainty of risk on the part of providers leading to withheld offers to participate [37]. In addition, patients who do agree to participate in clinical trials tend to be wealthier, have greater access to healthcare resources, be members of ethnic majorities, and often unrepresentative of overall populations of patients suffering from a given condition [33, 35, 37, 38, 39, 40]. Beyond questions of generalizability, recruitment challenges also cause delays to clinical trials, with an estimated 86% of trials delayed between 1 and 6 months, and some for even longer [41, 42].

These challenges often have severe effects on the outcomes of clinical trials, and to a certain extent new treatments available to patients. These effects can include inadequate statistical analysis of outcomes, cost overruns, extended duration of trials, increased costs of new medications, and treatments that potentially do not exhibit expected beneficial outcomes in understudied populations. [43, 39, 44, 45].

2.3 The Case for Software in Matching Patients for Clinical Trials

While computer software and NLP alone can likely not solve many of these challenges, research suggests that in many cases they can add significant value, time- and cost-savings

toward trial recruitment [39, 42]. For example, Thadani *et al* found electronic screening methods to significantly reduce the burden of manual chart review in one study by approximately 81% [42]. Examining multiple clinical trials, Penberthy *et al* similarly found up to a 20-fold decrease in staff time spent reviewing eligible patient records by using electronic screening software [39]. More recently, Ni *et al* used a combination of NLP techniques and structured data analysis to screen for potential clinical trial candidates and compared the results to a gold standard data set reviewed by medical doctors. The authors found their highest performing methods achieved an approximate 90% workload reduction in chart review and 450% increase in trial screening efficiency [46].

Drag-and-drop tools web-based tools such as i2b2 [3] and Leaf [2] and are also capable of determining eligible patients for clinical trials. i2b2, shown in Figure 2.1, uses a web-based user interface with biomedical concepts represented hierarchically on the left-side of the screen. Concepts are dragged and dropped into boxes on the right to create queries. After a user clicks "Run Query", the i2b2 API analyzes the input concepts to generate a database query on the i2b2 database schema to find patients meeting the given criteria. After query completion, a count of patients found is shown to the user at the bottom of the screen.

A screenshot of the Leaf user interface is shown in Figure 2.2. Leaf was inspired by i2b2 but introduced a number of innovations, including data model agnostic query generation, dynamic data de-identification and secure direct data export to a REDCap instance [47]. The process of building a query is similar to that of i2b2 and shown in Figure 2.3.

While both Leaf and i2b2 are widely used for cohort discovery, as discussed in Chapter 1, these tools have shortcomings such as relatively steep learning curves and the inability to represented certain complex queries, such as nested Boolean logic. From a practical standpoint, both tools also have the shortcoming that the number of boxes able to be used to represent queries is limited, a challenge that is particularly acute in the case of clinical trials eligibility criteria—which may run into tens of lines of criteria —are thus not representable as a single query in these tools. An alternative approach using NLP to analyze free-text criteria may potentially overcome these challenges.

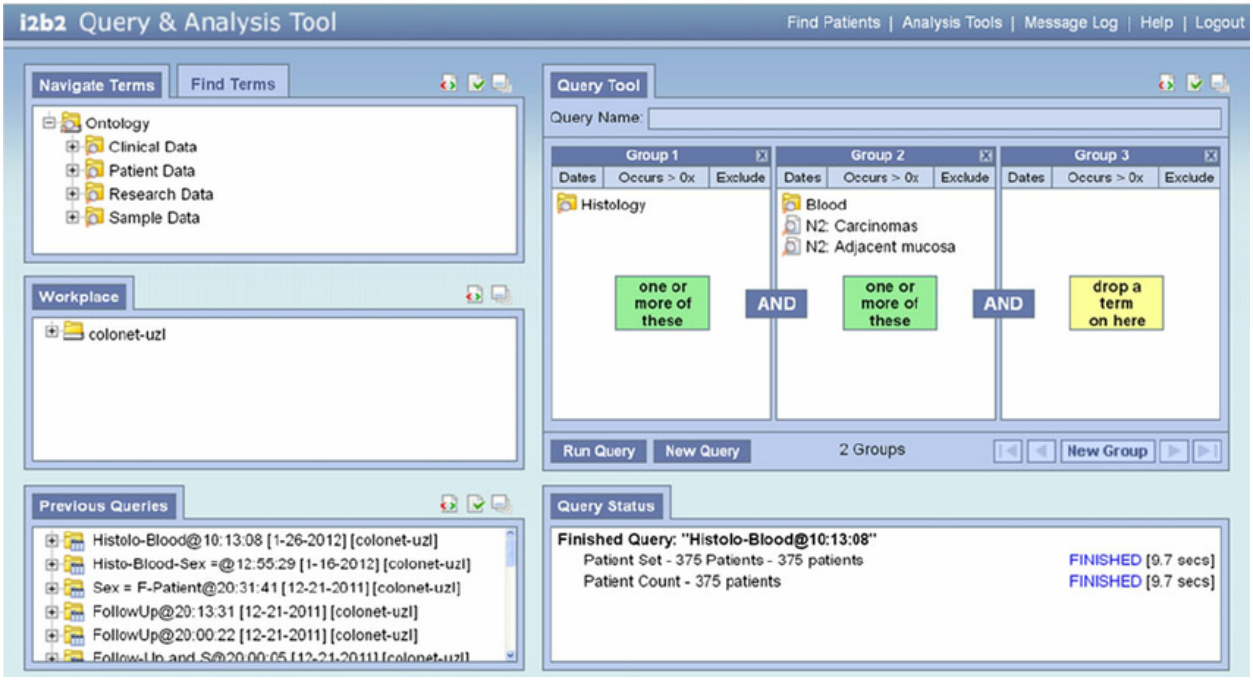


Figure 2.1: An example screenshot of the i2b2 user interface.

Though the aim of this project is to produce an application capable of general purpose cohort discovery—rather than solely for the purposes of clinical trial recruitment —clinical trials are a meaningful and valuable means by which to **gather data**, **evaluate performance**, and **measure potential real-world impact** of solutions for cohort discovery. Moreover, screening software and NLP have been demonstrated to dramatically improve trial recruitment efficiency in many scenarios. In terms of data, the website `https://clinicaltrials.gov`, maintained by the United States National Library of Medicine, hosts freely accessible descriptions of hundreds of thousands of clinical trials from around the world. Because clinical trials enrollments are in many cases recorded in EHRs (which also include the same patients' clinical data), they also can serve as a uniquely objective means of measuring the effectiveness of NLP systems in matching actual enrolled participants based on eligibility criteria. Put another way, an NLP-based system for matching patients to real-world eligibility criteria should reasonably be expected to find many or most patients

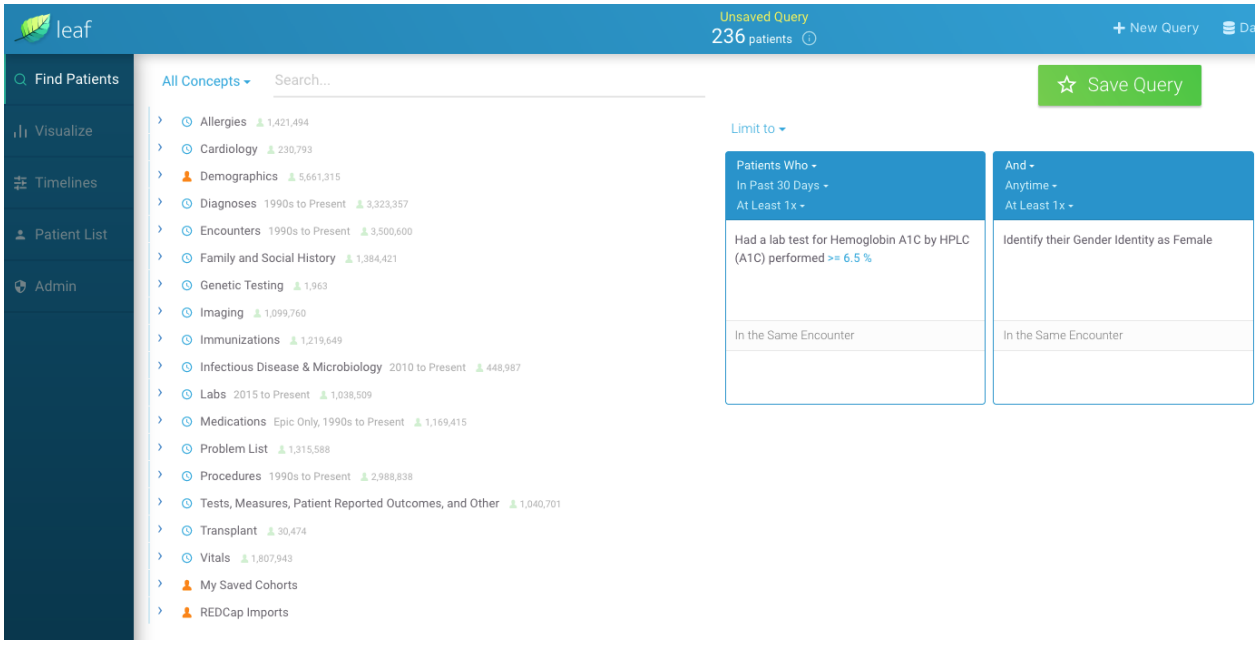


Figure 2.2: An example screenshot of the Leaf user interface.

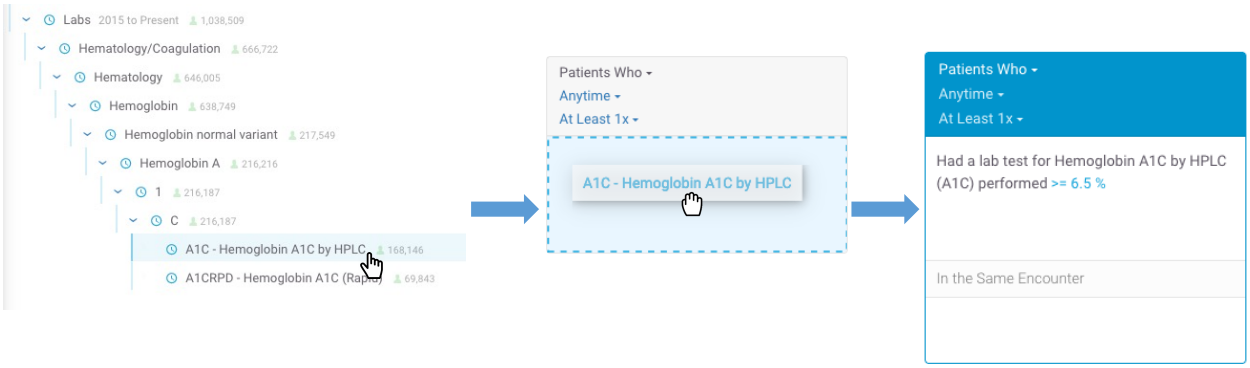


Figure 2.3: The Leaf query creation process. A user searches for relevant concepts on the left, then drags and drops them into boxes on the right.

enrolled in a given clinical trial - with the assumption that patients enrolled in those trials correctly met the necessary criteria as determined by study investigators. Though imperfect (e.g., a lack of diagnosis data for an existing condition may cause certain patients to be

inappropriately deemed ineligible), clinical trials are well-suited for evaluation of NLP-based cohort discovery systems and thus a focus of much of this project.

2.4 Challenges in Electronic Screening and NLP in Clinical Trials

Using NLP to determine patients potentially eligible for a clinical trial has numerous challenges. Consider, for example, a list of eligibility criteria such as[1]:

1. *Newly diagnosed with breast cancer and scheduled for surgery*
2. *18 years or above*
3. *Those who experience high psychological stress will enter the RCT whereas those with low stress will be followed in an observational questionnaire study*
4. *No severe psychiatric disease requiring treatment, e.g., schizophrenia*

While perhaps appearing deceptively simple, this example demonstrates many of the difficulties of this task. In criterion 1, "Newly" in "Newly diagnosed with breast cancer", suggests that diagnoses occurring further in the past (though how far is unclear) should not be included. Meanwhile, "surgery" in "scheduled for surgery" likely refers to surgery in relation to breast cancer, though this is not explicitly stated. In criterion 2, "18 years" likely refers to participants' age, but this too is not explicitly stated. Criterion 3, meanwhile, is a description of processes which will take place during the trial, but is not actually an eligibility criterion (i.e., participants may be eligible whether their actual stress levels are high or low). In criterion 4, "psychiatric disease" is non-specific and may refer to a large number of unstated conditions, aside from schizophrenia which is given as an example.

Appropriately interpreting the semantics and unstated requirements of these criteria can be challenging for both humans (such as study coordinators and recruiters) and NLP systems. For example, an NLP system may correctly determine that "breast cancer" refers

[1]Adapted from `https://clinicaltrials.gov/ct2/show/NCT03254875`

to a condition and "surgery" refers to a procedure, but may still fail if "Newly" is not determined to refer to breast cancer or to mean something occurring for the first time. In a subsequent step, an NLP system may normalize (i.e., determine a coded representation of a concept, for example a Unified Medical Language System [UMLS] code) "breast cancer" incorrectly as "Malignant Neoplasms" (C0006826) rather than "Malignant neoplasm of breast" (C0006142). In criterion 3, an NLP system may attempt to limit eligibility to patients with high stress levels, despite the criterion not being a formal restriction as such. In criterion 4, an NLP system may fail to reason that other unstated conditions, such as hallucinations, should also be excluded.

Beyond challenges in interpreting eligibility criteria, certain criteria may simply be absent or even incorrect in the data source the NLP system generates a query for. For example, Eastern Cooperative Oncology (ECOG) performance status scores [48] are frequently listed in eligibility criteria but often absent in structured clinical databases.

Last, an NLP system capable of identifying patients eligible for clinical trials should also be able to explain *why* patients are eligible and *what it did* to determine eligibility. This step is key to gaining user trust in the system [49, 50], but also challenging as many systems tend to prioritize performance over interpretability.

2.5 NLP Methods for Clinical Trial Recruitment

Various methods for matching eligibility criteria to cohorts of patients using NLP have been put forth by the research community [6, 7, 8, 9, 10, 11, 12, 13, 14]. NLP-based cohort discovery methods hold unique potential and appeal, as they are theoretically able to leverage existing eligibility criteria described in natural language, a medium researchers and investigators already use and are comfortable with. Recent methods which utilize NLP in some form can generally be grouped into 5 categories:

1. **Database query generation** in Structured Query Language (SQL) or similar systems using either (a) rules, (b) neural network-based encoder-decoder architectures, or both.

2. **Document ranking and classification** using clinical notes in terms of relevancy vis-à-vis a given eligibility criterion.

3. **Projection into embeddings** of patient medical history and trial eligibility criteria in a shared vector space and matching via similarity measurement or entailment.

4. **Logical representations and reasoning** to represent eligibility criteria and patient records, matching by combinations of Semantic Web technologies, ontologies, Description Logics, and rule-based reasoning.

5. A combination of the above.

Database query generation - SQL-based relational databases are widely used both commercially and within academic institutions, and as such SQL is perhaps unsurprisingly often the target language in natural language to database query research [51]. Yuan *et al* developed Criteria2Query (shown in Figure 2.4), a hybrid information extraction (IE) pipeline and application which uses both rules and machine learning to generate database queries on an Observational Medical Outcomes (OMOP) [52] database. This work was expanded by Fang *et al*, who added functionality for iterative query generation via human correction and adjustment [8]. Although not specific to RCTs, other highly relevant recent work on query generation in the biomedical domain has been done using Encoder-Decoder neural architectures for transforming clinical natural language questions into SQL queries [53, 54, 55, 56, 12]. Park *et al* [54] experimented with transforming medical questions generated in the MIMICSQL data set [57, 55] using both SQL and SPARQL queries with varying database schema representations. Bae *et al* similarly experimented with methods for handling typos, misspellings, and abbreviations in generating SQL queries from natural language questions. Pan *et al* [56] leveraged intermediate abstract syntax tree-based representations and a SQL grammar-based Decoder architecture for dynamic database schema matching.

Document ranking and classification - Focusing on clinical notes, Chen *et al* [10] used hybrid rule-based heuristics and sentence pattern-matching to detect criteria structure,

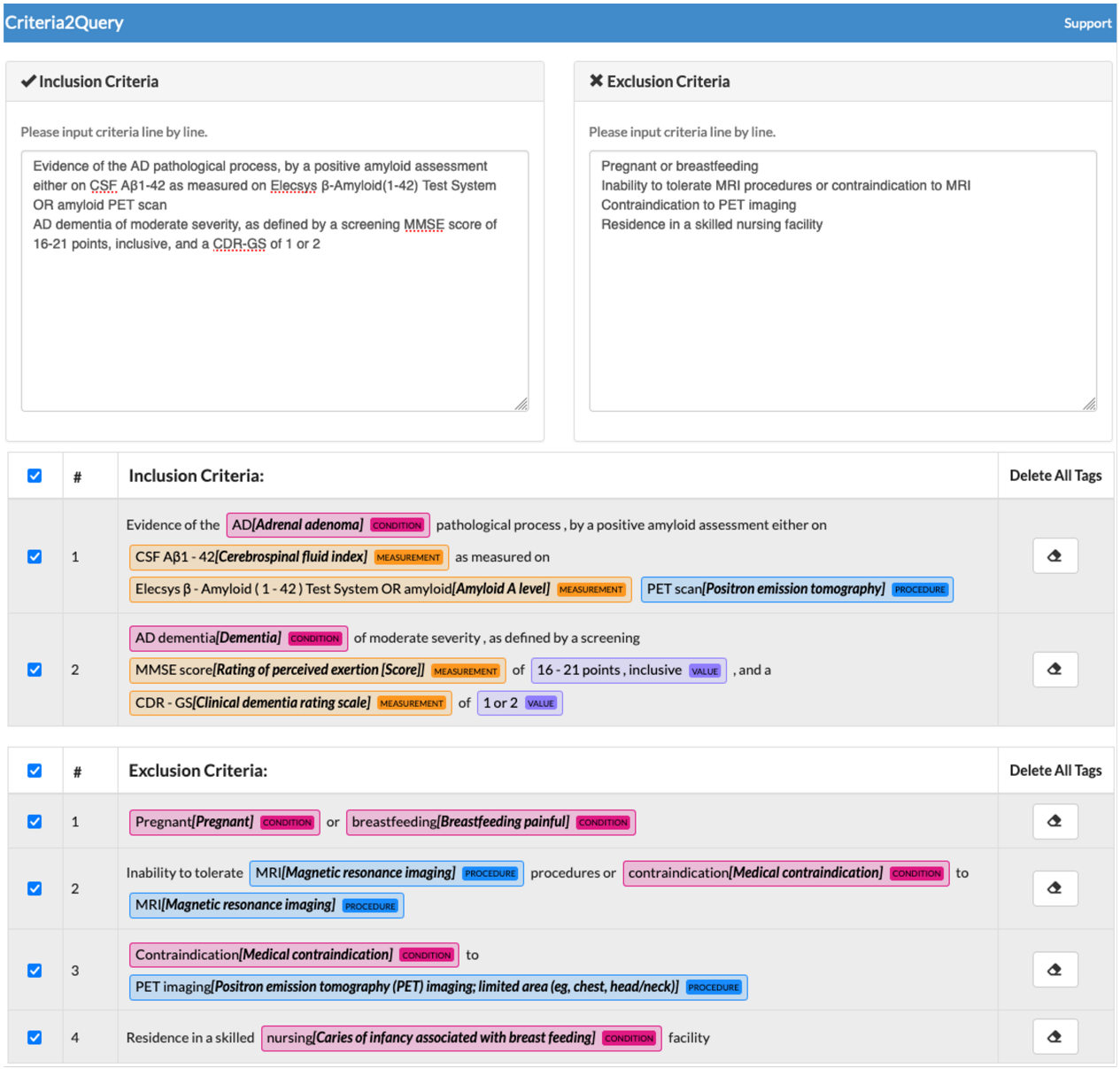


Figure 2.4: An example screenshot of the Criteria2Query application. Users type eligibility criteria in free text or enter a clinical trials "NCT" identifier. The application then generates a SQL query and executes it against an OMOP database in order to identify eligible patients.

as well as a combination of neural network-based bi-directional long short-term and conditional random field (biLSTM+CRF) architecture and knowledge graphs using the UMLS for

determining condition, lab, procedure and drug relationships. Soni and Roberts [7] utilized the BERT Transformer architecture [58] and Lucene [59] to summarize, rank and classify clinical notes as relevant to a given eligibility criterion, with the most relevant notes predicted to be eligible.

Embedding projections - Dhayne *et al* [12] experimented with treating patient-to-RCT matching as a joint embedding and similarity measurement problem while also incorporating the SNOMED-CT ontology to infer basic "is-a" and "has-type" relations between concepts. Similarly, Zhang *et al* [9] used joint patient and eligibility criteria embeddings for entailment prediction, where predicting that a patient can be inferred from a given eligibility criteria equates to eligibility.

Logical representations and reasoning - Patrao *et al* developed Recruit [11], an ontology-driven trial recruitment system which transformed SQL relational data to Resource Description Framework (RDF) graph-based triples. The RDF triples in turn were made query-able by use of an OWL-based reasoning system [60] and normalization techniques to infer cancer staging. Building upon earlier work [61, 62, 63], Baader *et al* [64] explored the use of Description Logics and ontologies in matching patients in the MIMIC data set to logical representations of eligibility criteria, for example representing "Diabetes mellitus type 1" as "$\exists_y$.diagnosed_with(x, y) $\wedge$ Diabetes_mellitus_type_1(y)". Liu *et al* [13] used domain experts to manually translate criteria into a custom syntax parsable by software. For example, the criterion "Patients more than 18 years old when they received the treatment" would be represented as "$\#Inclusionfeatures['StartDate'] >= demographics['BirthDate']+@YEARS(18)$". Parsed eligibility criteria were then executed on a proprietary database schema to determine eligible patients.

2.6 Corpora

A key element for any NLP-based approach in identifying patients is robust corpora which capture eligibility criteria semantics sufficiently for high-accuracy query generation. Such corpora can serve as reliable benchmarks for purposes of comparing NLP methods as well

as training data sets. A number of corpora have been published. Weng *et al* created EliXR [65], a pipeline of rule-based methods for syntactic parsing of eligibility criteria using pattern matching. EliXR was validated using a subset of 1,000 randomly selected eligibility criteria annotated by human annotators and compared to syntax trees extracted by their system. This work was expanded by Boland *et al* to support normalization of complex temporal patterns [66]. Neither data set was publicly released. Kang *et al* developed EliIE [67], a hybrid machine-learning and rule-based system for eligibility criteria extraction and normalization. EliIE was evaluated against a human-annotated corpus of eligibility criteria from 230 trials related to Alzheimer's Disease. EliIE was the first system for eligibility criteria extraction which used machine-learning methods (a Conditional Random Field, or CRF [68]) and the first to use an extraction schema based on the Observational Medical Outcomes Partnership (OMOP) [52] common data model. The EliIE corpus was similarly not publicly released, and also limited in terms of generalizability by focusing on Alzheimer's Disease alone.

More recently, Kury *et al* released Chia [69], a human-annotated corpus of eligibility criteria from 1,000 Phase IV clinical trials across various disease domains. Like EliIE, Chia used an annotation schema based on the OMOP common data model. Following Chia, Sun *et al* created a corpus of COVID-19-related trials' eligibility criteria [70]. The authors used a similar annotation schema to that of Chia, but included separate annotation layers for cohorts, criterion, named entities, and concepts, which they referred to as a hierarchical annotation model.

Yu *et al* [71] released a corpus designed for direct text-to-query generation with semantic parsing, however given the relative simplicity of generated queries to date compared to the complexity of clinical databases, it's not clear this approach is yet viable for real-world clinical trials recruitment.

2.7 Clinical Knowledge Bases

A system capable of reasoning on non-specific eligibility criteria necessarily must store some representation of clinical phenomena and their relations. Such information is often described

as a knowledge base (KB) [72], or an information store often composed of questions and answers, entities and relations, or factoids. Large-scale KBs, such as DBpedia [73] are represented as graphs of tuples, where a tuple represents two entities and a relation between them (also often called nodes and edges), such as { "asthma", "is-a", "disorder of the respiratory system" }. Chains of tuple-based entities joined by relations can therefore represent fairly complex phenomena in a compositional fashion. In medicine, the most widely used KB is the Unified Medical Language System, or UMLS [19]. The UMLS is a metathesaurus of controlled vocabularies and terminologies such as ICD-9, ICD-10, LOINC, SNOMED, CPT, and others, with mappings between them. While often accessed as a relational database, the UMLS can also be transformed and represented tuple form [74].

Within the clinical domain, KBs (often including the UMLS) have been used for various purposes related to retrieval, reasoning, disambiguation, and question answering. For example, Martinez *et al* and Sankhavara *et al* demonstrated use of the UMLS for query expansion, enabling patient record retrieval via use of synonyms [75, 76]. For reasoning, Shulz and Hahn demonstrated a technique for leveraging the UMLS to reasoning upon anatomical structure [77]. Kazi *et al* demonstrated the use of the UMLS for basic question-answering of medical students [78].

More recent work has explored alternative representations of KBs within neural networks. Lin *et al* explored neural network-based inference techniques for inferring missing relations among entities based on other information present in the UMLS and other KBs [79]. Hao *et al* and Huang *et al* examined approaches for integrating the UMLS into pre-training of BERT [58] embeddings [80, 81]. Petroni *et al* point out that even without fine-tuning for a specific domain or task, pre-training of language models with vast data from the internet exhibit reasonable performance on many tasks which necessitate basic reasoning [82].

2.8 Summary

This chapter discussed the importance of clinical trials to biomedical research and human health, as well as challenges in cost and recruitment. We next discussed motivation for and

challenges in the use of NLP in patient matching for clinical trials, as well as related research in clinical KBs and published corpora related to clinical trials.

Chapter 3

LEAF CLINICAL TRIALS CORPUS

3.1 Overview

The NLP tasks involved in transforming eligibility criteria into database queries may include **named entity recognition** (NER) to tag meaningful spans of text as named entities, **relation extraction** to classify relations between named entities, **normalization** to map named entities to common coded representations (e.g., ICD-10), **negation detection** to detect negated statements (e.g., "not hypertensive") and so on. Gold standard corpora quality can thus directly affect performance and the validation of each of these tasks. Such corpora can serve as reliable benchmarks for purposes of comparing NLP methods as well as training data sets.

In this chapter we describe the creation of a gold standard corpus of human-annotated clinical trial eligibility criteria and performance data measured during our evaluation. Predictive models trained on this corpus enable and are used in subsequent aims of this project. Section Section 3.2 discusses motivations and need for the corpus. Section 3.3 describes the LCT annotation schema. We discuss our evaluation methods in Section 3.4, limitations in Section 3.5, and provide a summary in Section 3.6. This work was published as Dobbins *et al* in 2022 [18].

3.2 Motivation

We aimed to develop an expressive, task-oriented annotation schema which could capture a wide range of medical concepts and logical constructs present in eligibility criteria. To accomplish this, we first analyzed previously published corpora [65, 66, 67, 69] and expanded the list of included biomedical phenomena to fully capture the context and logic present in

real clinical trials criteria. As one example, we introduced an entity called *Contraindication* to reflect where use of a given treatment is inadvisable due to possible harm to the patient.

Figure 3.1 illustrates the importance of corpora structure for the task of query generation, using examples of eligibility criteria annotated using our annotation schema and corresponding hypothetical Structured Query Language (SQL) queries. In the first eligibility criterion, "preeclampsia" is explicitly named, and thus can be directly normalized to an ICD-10 or other coded representation. However, eligibility criteria involving non-specific drugs, conditions, procedures, contraindications, and so on are used frequently in clinical trials. In the second criterion in Figure 3.1, "diseases" in "diseases that affect respiratory function" is non-specific, and must be reasoned upon in order to determine appropriate codes, such as asthma, chronic obstructive pulmonary disease (COPD), or emphysema. Programmatically reasoning to generate queries in such cases would be challenging and often impossible if the underlying semantics were not captured appropriately. With this in mind, we developed the Leaf Clinical Trials, or LCT, annotation schema in order to enable reasoning and ease query generation for real-world clinical trials use. As the second example in Figure 3.1 shows, the LCT annotation captures the semantics of complex criteria, with changes to "respiratory function" annotated using a *Stability[change]* entity and *Stability* relation, and the cause, "diseases" annotated with a *Caused-By* relation.

3.3 Annotation schema

The LCT annotation schema is designed with the following goals and assumptions:

1. The annotation schema should be **practical** and **task-oriented** with a focus on facilitating ease of query generation.

2. A greater number of **more specific**, **less ambiguous** annotated phenomena should be favored over a smaller number of possibly ambiguous ones.

3. Annotations should be **easily transformable** into composable, interconnected pro-

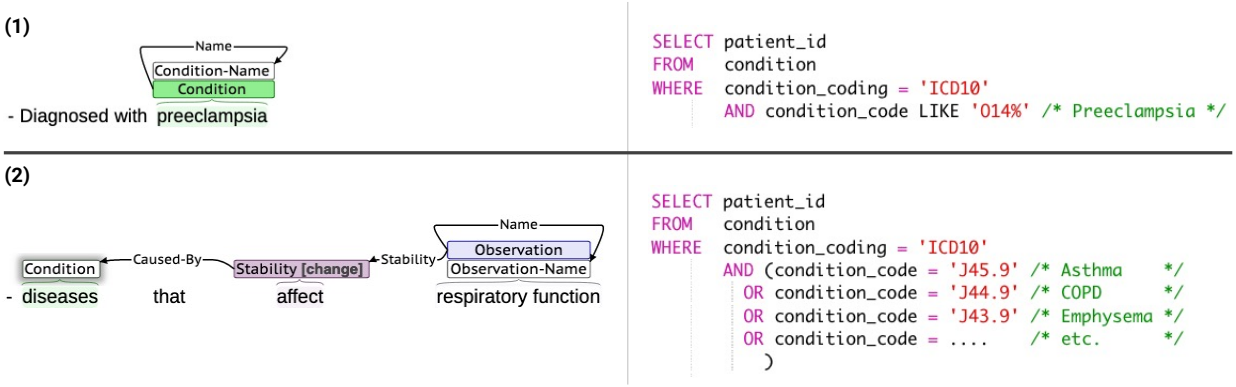


Figure 3.1: Example eligibility criteria annotated used the LCT corpus annotation schema (left) and corresponding example SQL queries (right) using a hypothetical database table and columns. Annotations were done using the Brat annotation tool. The ICD-10 codes shown are examples and not intended to be exhaustive.

grammatic objects, trees, or node-edge graph representations.

4. The annotation schema should **model eligibility criteria intent and semantics** as closely as possible in order to ensure generated queries can do the same.

The LCT annotation schema is composed of **entities** and **relations**. Entities refer to biomedical, demographic, or other named entities relevant to eligibility criteria, and are annotated as a span of one or more tokens. We organized LCT entities into the following categories:

- **Clinical** - *Allergy, Condition, Condition-Type, Code, Contraindication, Drug, Encounter, Indication, Immunization, Observation, Organism, Specimen, Procedure, Provider.*
- **Demographic** - *Age, Birth, Death, Ethnicity, Family-Member, Language, Life-Stage-And-Gender.*
- **Logical** - *Exception, Negation.*

- **Qualifiers** - *Acuteness, Assertion, Modifier, Polarity, Risk, Severity, Stability.*

- **Comparative** - *Criteria-Count, Eq-Comparison* (an abbreviation of "Equality Comparison"), *Eq-Operator, Eq-Temporal-Period, Eq-Temporal-Recency, Eq-Temporal-Unit, Eq-Unit, Eq-Value.*

- **Other** - *Coreference, Insurance, Location, Other, Study.*

The LCT corpus also includes 7 *Name* entities: *Allergy-Name*, *Condition-Name*, *Drug-Name*, *Immunization-Name*, *Observation-Name*, *Organism-Name* and *Procedure-Name*. *Name* entities serve a special purpose in the LCT corpus, as they indicate that a span of text refers to a *specific* condition, drug, etc., as opposed to *any* condition or drug. *Name* entities overlap with their respective general entities. For example, the span "preeclampsia" refers to a specific condition, and would thus be annotated as both a *Condition* and *Condition-Name*, while the span "diseases" is non-specific and would be annotated as only *Condition*. A full listing of the LCT annotation guidelines can be found at `https://github.com/uw-bionlp/clinical-trials-gov-annotation/wiki`.

We defined a total of 50 entities in the LCT corpus. Examples of selected representative entities are presented in Table 3.1. In our representation, a subset of entities have **values** as well. For example, an *Encounter* may have a value of *emergency*, *outpatient* or *inpatient*. Values are optional in some entities (such as *Encounters* or *Family-Member*, where they may not always be clear or are intentionally broad) and always present in others. In the example annotations presented below, values are denoted using brackets ("[...]") following entity labels.

Relations serve as semantically meaningful connections between entities, such as when one entity acts upon, is found by, caused by, or related in some way to another. We categorize relations into the following:

- **Alternatives and Examples** - *Abbrev-Of, Equivalent-To, Example-Of.*

Category	Entity	Values	Example Text
Clinical	Condition	–	Diagnosed with hypertension$_{Condition}$ in past year
	Contraindication	–	any contraindications$_{Contraindication}$ to vaginal delivery
	Drug	–	on beta blockers$_{Drug}$
	Encounter	emergency, outpatient, inpatient	recently admitted$_{Encounter[inpatient]}$ to a hospital
	Observation	lab, vital, clinical-score, survey, social-habit	Platelet count$_{Observation[lab]}$ less than 500
	Procedure	–	Undergoing or scheduled for a colonoscopy$_{Procedure}$
Demographic	Age	–	43 years old$_{Age}$
	Birth	–	born$_{Birth}$ within the past 6 months
	Family-Member	mother, father, sibling, etc.	history of maternal$_{Family-Member[mother]}$ breast cancer
	Language	–	Speaks English$_{Language}$ or Spanish$_{Language}$
Logical	Negation	–	with no$_{Negation}$ systemic disease
Qualifier	Assertion	intention, hypothetical, possible	which may$_{Assertion[hypothetical]}$ cause conditions
	Modifier	–	alcohol$_{Modifier}$ or substance$_{Modifier}$ abuse
	Polarity	low, high, positive, negative	showing elevated$_{Polarity[high]}$ serum creatinine
	Risk	–	at heightened potential$_{Risk}$ for suicide
	Severity	mild, moderate, severe	with serious$_{Severity[severe]}$ complications from surgery
	Stability	stable, change	conditions known to affect$_{Stability[change]}$ mood
Comparative	Criteria-Count	–	at least 3 of the following conditions:$_{Criteria-Count}$
	Eq-Comparison	–	greater than 50ml$_{Eq-Comparison}$
	Eq-Temporal-Period	past, present, future	Active$_{Eq-Temporal-Period[present]}$ illness
	Eq-Temporal-Recency	first-time, most-recent	Latest$_{Eq-Temporal-Recency[most-recent]}$ BMI
Other	Location	residence, clinic, hospital, unit, emergency-department	Seen at diabetes care clinic$_{Location[clinic]}$

Table 3.1: **Examples of representative LCT annotation schema entities.** A full listing of all entities can be found in the LCT annotation guidelines at `https://github.com/uw-bionlp/clinical-trials-gov-annotation/wiki`.

- **Clinical** - *Code, Contraindicates, Indication-For, Name, Provider, Specimen, Stage, Type.*
- **Dependent** - *Caused-By, Found-By, Treatment-For, Using.*
- **Logical** - *And, If-Then, Negates, Or.*
- **Qualifier** - *Acuteness, Asserted, Dose, Modifies, Polarity, Risk-For, Severity, Stability.*
- **Comparative** - *After, Before, Criteria, Duration, During, Max-Value, Min-Value, Minimum-Count, Numeric-Filter, Operator, Per, Temporal-Period, Temporal-Recency, Temporal-Unit, Temporality, Unit, Value.*
- **Other** - *From, Except, Has, Is-Other, Location, Refers-To, Study-Of.*

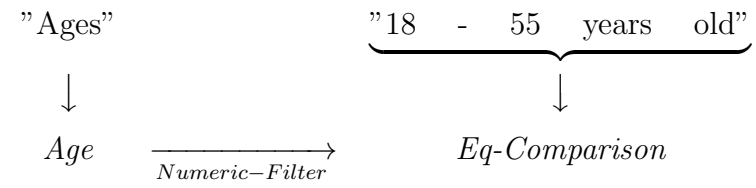
We defined a total of 51 relations in the LCT corpus. Examples of relation criteria are shown in Table 3.2.

In our annotations, some entity spans overlap with other entity spans in order to fully capture complex underlying semantics. Consider for example, the expression "Ages 18-55 years old". While an *Age* entity may be assigned to token "Ages", if an *Eq-Comparison* entity alone were assigned to the span "18-55 years old", the underlying semantics of the tokens "18", "-", "55", and "years" would be lost. In the following examples, we use the term **fine-grained entity** to refer to entities which are sub-spans of other **general entities**. Fine-grained entities are linked to general entities by relations. We use down arrow symbols ($\downarrow$) to denote entity annotation and left and right arrow symbols ($\leftarrow$ and $\rightarrow$) to denote relations. The ($+$) symbols denote overlapping entities on the same span.

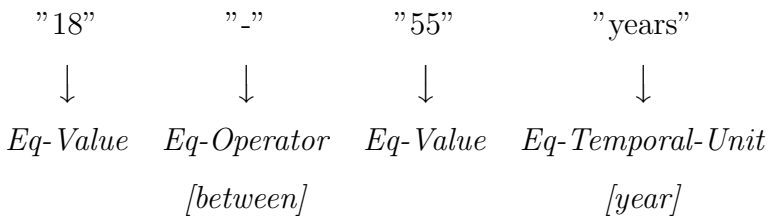
The example expression "Ages 18-55 years old" would be annotated in three layers. In the first layer, the expression is annotated with *Age* and *Eq-Comparison* general entities with a relation between them:

Category	Relation	Example Annotation
Alternatives and Examples	Abbrev-Of	Post Concussion Syndrome $\xleftarrow[Abbrev-Of]{}$ (PCS) (Condition / Condition)
	Equivalent-To	Thrombocytopenia $\xleftarrow[Equivalent-To]{}$ platelets > 100,000/mm3" (Condition / Observation[lab])
	Example-Of	skin condition $\xleftarrow[Example-Of]{}$ (e.g. eczema)" (Condition / Condition)
Clinical	Contraindicates	conditions contraindicating $\xrightarrow[Contraindicates]{}$ MRI (Contraindication / Procedure)
Dependent	Caused-By	swellings $\xrightarrow[Caused-By]{}$ due to trauma (Observation / Condition)
	Found-By	lesion $\xrightarrow[Found-By]{}$ seen on standard imaging (Observation / Procedure)
	Treatment-For	coronary bypass surgery $\xrightarrow[Treatment-For]{}$ for atherosclerosis (Procedure / Condition)
	Using	total knee arthroplasty $\xrightarrow[Using]{}$ with spinal anesthesia (Procedure / Procedure)
Logical	If-Then	BMI greater than 38 $\xleftarrow[If-Then]{}$ for women (Eq−Comparison / Life−Stage−And−Gender[female])
Qualifier	Risk-For	risk $\xrightarrow[Risk-For]{}$ of death (Risk / Death)
	Severity	mild $\xleftarrow[Severity]{}$ symptoms (Severity[mild] / Observation)
	Stability	hemodynamically $\xrightarrow[Stability]{}$ unstable (Observation / Stability[change])
Temporal and Comparative	After	infected $\xrightarrow[After]{}$ following admission (Condition / Encounter[inpatient])
	Before	diagnosisof aortic stenosis $\xrightarrow[Before]{}$ prior to visit (Condition / Encounter)
	Duration	type 1 diabetes $\xrightarrow[Duration]{}$ for at least 1 year (Condition / Eq−Comparison)
	During	mechanically ventilated $\xrightarrow[During]{}$ while admitted (Procedure / Encounter[inpatient])
	Numeric-Filter	body weight $\xrightarrow[Numeric-Filter]{}$ less than 110 pounds (Observation[vital] / Eq−Comparison)
	Minimum-Count	admitted $\xrightarrow[Minimum-Count]{}$ at least twice (Encounter[inpatient] / Eq−Comparison)
	Temporality	seen $\xrightarrow[Temporality]{}$ within past 6 months (Encounter / Eq−Comparison)
Other	Location	admitted $\xrightarrow[Location]{}$ to the ICU (Encounter[inpatient] / Location[unit])

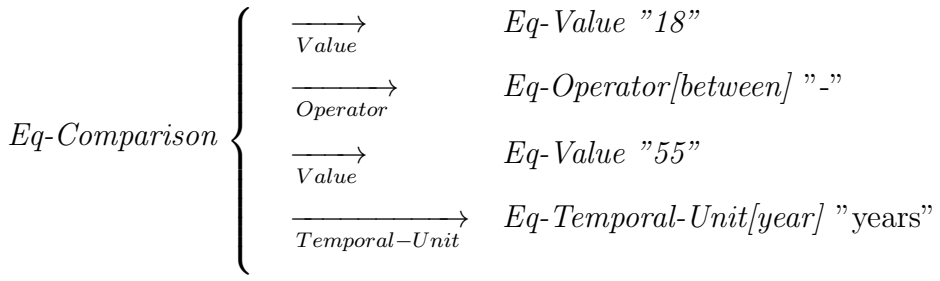
Table 3.2: **Examples of representative relations.** Direction of arrows indicates role, i.e., subject → target entity.



In the second layer, fine-grained entities with respective values are annotated:



In the third layer, relations connecting fine-grained entities to the general *Eq-Comparison* entity are added:



This multilayered annotation strategy allows significant flexibility in capturing entities and relations in a slot-filling fashion, simplifying the task of downstream query generation.

The LCT annotation schema contributes the following novel features: (1) deep granularity in entities and relations, which enables (2) rich semantic representation, closely capturing the intent of complex clinical trial eligibility criteria and facilitating accurate query generation.

3.3.1 Deep Entity and Relation Granularity

We assume that more specific annotation labels are generally more straightforward to generate accurate queries with. For example, within the span, "preceding six months", annotating the token "preceding" as *Temporal* (an entity type in Chia) may appear to be adequate, given that an English-speaking human would understand that this refers to the past. Without further information, however, a naïve algorithm would be unable to determine (1) whether such a entity refers to the past, present, or future, (2) that the token "six" refers to a numeric value, and (3) that "months" refers to a unit of temporal measurement. In such cases, most query generation algorithms introduce additional rule-based or syntactic parsing modules, such as SuTime [83] to further normalize the phrase to a value [65, 6]. This ambiguity in label semantics creates unnecessary complexity in downstream systems, requiring that the same text be processed a second time.

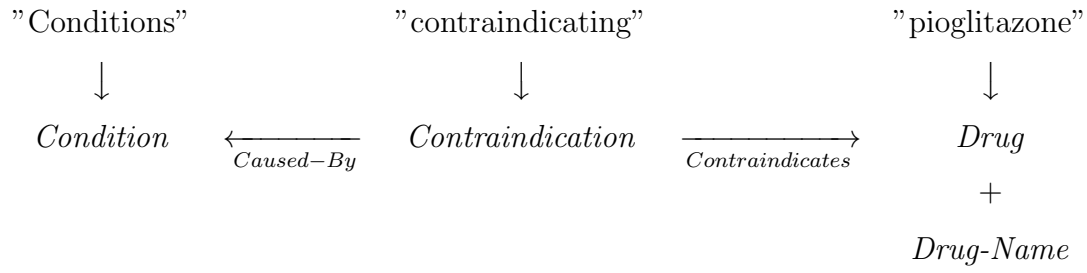
In contrast, we designed the LCT annotation schema to favor discrete, explicit entities and relations where possible, with an aim toward reducing the need for additional normalization steps needed for query generation. In our annotation schema, this example would be annotated with the following fine-grained entities:

"preceding"	"six"	"months"
↓	↓	↓
Eq-Temporal-Period	*Eq-Value*	*Eq-Temporal-Unit*
[past]		*[month]*

As shown in the example, each token is uniquely annotated, with the values *[past]* and *[month]* serving to more clearly disambiguate semantics of the temporal phrase to a normalized temporal value. Moreover, as fine-grained entities are connected by relations to general entities, which can in turn have relations to other general entities, the LCT annotation schema is able to capture eligibility criteria semantics at a deeper level than other corpora.

3.3.2 Rich Semantic Representation

Certain eligibility criteria cannot be directly translated into queries, but instead must first be reasoned upon. For example, a query to find patients meeting the criterion of "conditions contraindicating pioglitazone" requires reasoning to first answer the question, *What* conditions contraindicate use of pioglitazone? Such reasoning may be performed by a knowledge base or other methods, but cannot be done unless the contraindicative relation is first detected:



As the span "Conditions" is labeled *Condition* but does not have an overlapping *Condition-Name* entity, it is considered unnamed and thus would need to be reasoned upon to determine. "[P]ioglitazone", on the other hand, includes a *Drug-Name* entity and is thus considered named. The absence of overlapping *Name* entities serves as an indicator to downstream applications that reasoning may be needed to determine relevant conditions or drugs.

3.3.3 Comparison to Chia

We designed the LCT annotation schema by building upon the important previous work of EliIE and Chia. Chia builds upon EliIE and is more recent. Figure 3.2 shows comparison examples of annotations of the same eligibility criteria using the two corpora in the Brat annotation tool [84].

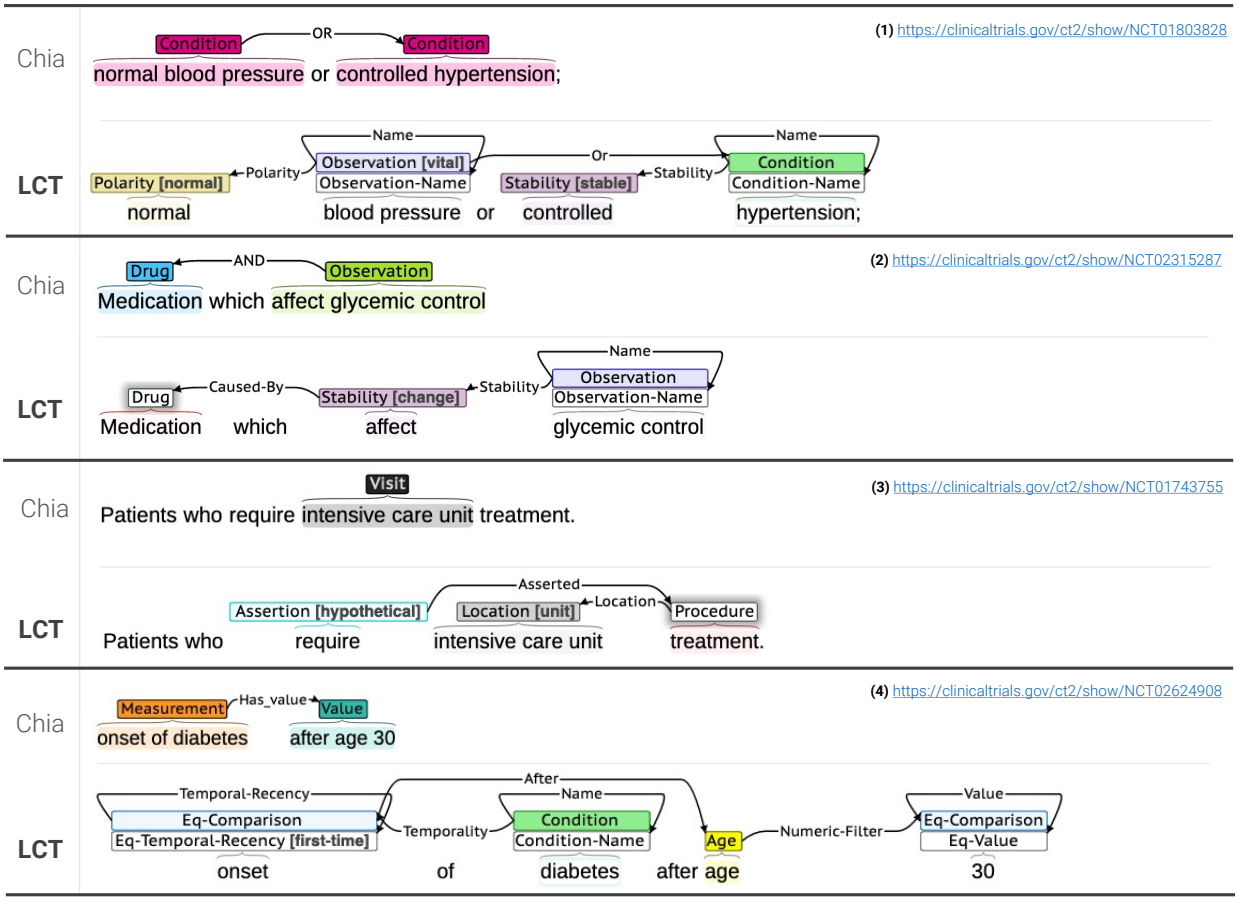


Figure 3.2: Examples of clinical trials eligibility criteria annotated with Chia and LCT annotation schemas. Each example shows a criterion from a Chia annotation (above) and an LCT annotation of the same text for purposes of comparison (below).

Capturing Entity Semantics

Example 2 of Figure 3.2 demonstrates the need to closely capture semantics in clinical trials eligibility criteria for unnamed entities. The span "Medication" in "Medication[s] which affect glycemic control" refers to *any* drug which potentially affects glycemic control. As discussed, the LCT annotation schema uses *Name* entities to handle such cases, where the absence in this example of a *Drug-Name* entity indicates that "Medication" refers to any

drug, and thus may need to be determined by downstream use of a knowledge base or other methods.

As can be seen, Chia does not differentiate between named and unspecified drugs, conditions, procedures and so on. While it is true that for query generation one may need to normalize these spans to coded representations (e.g., ICD-10, RxNorm or LOINC codes) and may in the process find that the span "Medication" is not a particular medication (and thus can be assumed to be *any* medication), such a workaround nonetheless complicates usage of the corpus in finding and handling such cases in a more direct, less error-prone way.

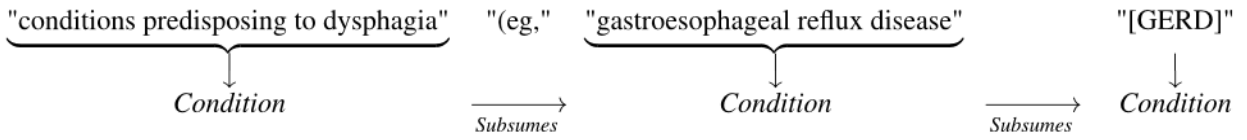
Consider also the phrase, "after age 30" in example 4 in Figure 3.2. In Chia, disambiguation of time units, values, and chronological tense must be performed by additional processing, as the Chia *Value* entity provides no information as to the semantics of the component sub-spans. In contrast, in the LCT corpus the tokens "after", "age", "30" are annotated with the explicit entities and relations to enable more straightforward query generation.

Capturing Relation Semantics

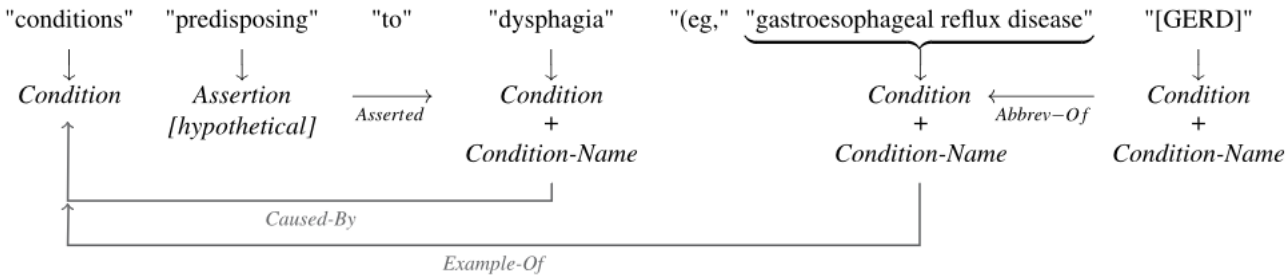
Eligibility criteria frequently contain entities which relate to other entities in the form of examples, abbreviations, equivalencies, or explicit lists. Chia uses *Subsumes* relations to denote that one or more entities are a subset, depend on, or are affected by another entity in some way. In many cases however these entities leave significant semantic ambiguity which may complicate query generation. Consider the phrase:

"conditions predisposing to dysphagia (eg, gastroesophageal reflux disease [GERD] ...)"

In this case, "gastroesophageal reflux disease" is an *example* of a condition, while "GERD" is an *abbreviation*. However, both are *Subsumes* relations in the Chia annotation:

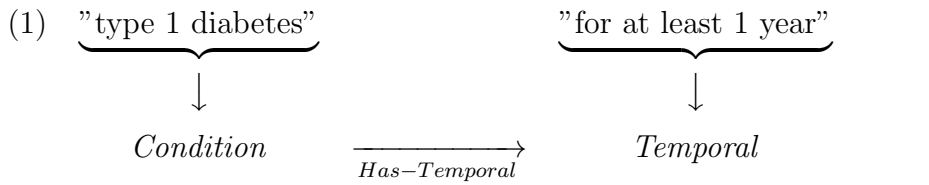


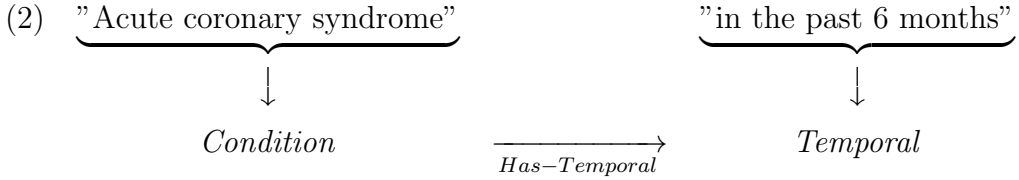
In contrast, in the LCT annotation schema this example would be annotated as:



The LCT annotation uses *Abbrev-Of* and *Example-Of* relations to clearly differentiate relations between "gastroesophageal reflux disease", "GERD", and "conditions predisposing to dysphagia". Additionally, rather than grouping the latter into a single *Condition* entity, the LCT is also much more granular, with the annotation reflecting that dysphagia is a condition patients are hypothetically predisposed to (due to other conditions such as GERD), but not necessarily actively afflicted by.

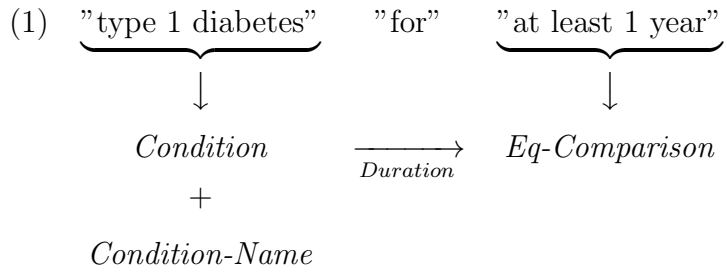
Another example illustrating the importance of capturing relation semantics can be seen in the following Chia annotations:

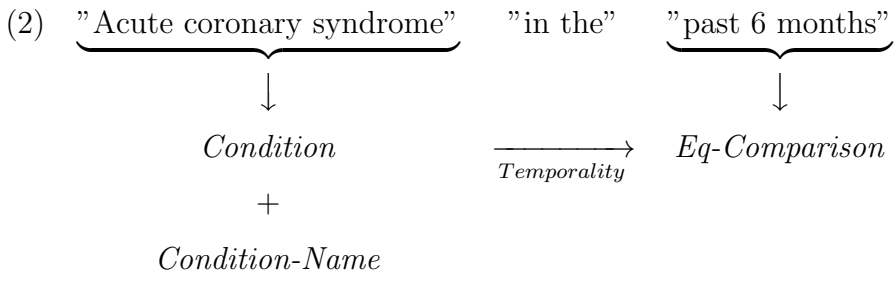




While syntactically similar, the semantics in chronology expressed in the two criteria are different. In (1), "type 1 diabetes for at least 1 year" suggests that the diagnosis of type 1 diabetes mellitus should have occurred at least 1 year prior to the present. In other words, a unit of temporal measurement (1 year), should have passed since initial diagnosis. In contrast, "Acute coronary syndrome in the past 6 months" (2) suggests that a range of dates between the present and a past event (past 6 months), should have passed since the diagnosis. In Chia, however, the same *Has-Temporal* relation is used for both, blurring distinctions between *durations of time* versus *ranges of dates*, potentially leading to errors during query generation.

In the LCT annotation schema, these would be annotated as (omitting fine-grained entities for brevity):





The LCT annotations distinguish these types of temporal semantics by using distinct *Duration* and *Temporality* relations, allowing downstream queries to more accurately reflect researcher intent. The LCT corpus also does not include "for" or "in the" as part of the entities.

Data Model Mapping

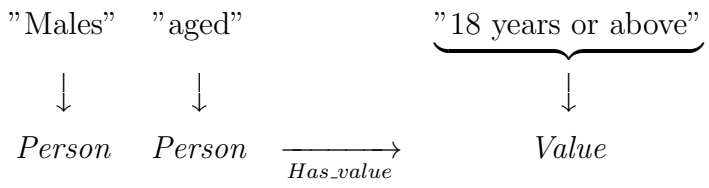
The Chia annotation schema is mapped to the OMOP Common Data Model [52] and is designed to ease integration with other OMOP-related tools and generation of SQL queries on OMOP databases. Chia OMOP-derived entities generally follow the naming convention of OMOP domains and SQL database tables, such as *Person*, *Condition*, *Device* and so on.

The LCT annotation schema takes a different approach by intentionally avoiding direct mappings to data models. This approach was chosen to (1) allow the annotation entities and relations flexibility to be transformed to any data model (including but not limited to OMOP) and (2) provide flexibility in capturing criteria important to the task of query generation, even when such criteria are not represented in OMOP.

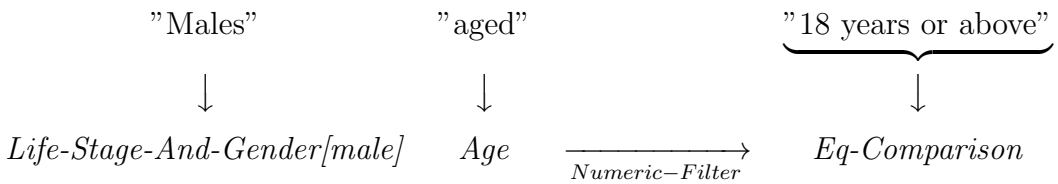
A disadvantage of directly coupling an annotation schema to a data model is evidenced by criteria such as:

"Males aged 18 years and above"

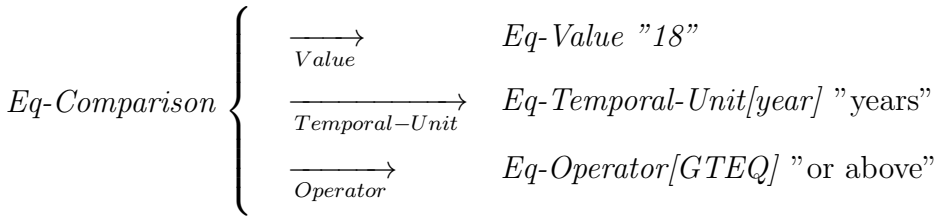
In Chia, spans related to gender and age share the same *Person* entity:



The use of the Chia *Person* entity across gender and age results in loss of information and complications for query generation. As with quantitative and temporal annotations, the generic *Person* entity again forces the burden of normalization and additional parsing to downstream applications. In contrast, this example would be annotated first using general entities and relations in the LCT annotation schema:



Followed by fine-grained entities and values:



The LCT annotation captures the male and age spans as distinguishable entities, closely preserving the semantics of the original text.

3.3.4 Annotation process

We extracted 1,020 randomly selected clinical trials eligibility descriptions from `https://clinicaltrials.gov` from 2018 to 2021, 20 for training and inter-annotator comparison and 1,000 for post-training annotation.

During annotation, 14 documents were found to be information poor (often with no spans to annotate) and discarded, resulting in 1,006 total annotated eligibility descriptions. Annotation was performed by two annotators, the first a biomedical informatician and the second a computer scientist. For initial annotation training, 20 documents were distributed to both annotators. Annotation was done in the following steps:

1. Annotation meetings were held bi-weekly for 3 months following initial annotation training in which the annotation guidelines were introduced. Initial meetings focused on discussion of annotation guideline implementation and revision.

2. After annotation guideline revisions and annotation training were completed, eligibility criteria were assigned to each annotator, with each clinical trial eligibility criteria annotated by a single annotator using the BRAT annotation tool [84]. Due to differences in time availability for annotation, roughly 90% (887 documents) of the annotation task was performed by the first annotator, and 99 documents by the second annotator.

3. At the point in which 50% of the corpus was annotated, we trained two neural networks (one for general entities and another for fine-grained entities) using the biLSTM+CRF-based NeuroNER tool [85] on our manually annotated eligibility criteria to predict annotations for the remaining 50%.

4. Manual annotation was completed on the remaining 50% of eligibility descriptions by editing and correcting the predicted entities from NeuroNER in (3).

The resulting corpus included 887 single-annotated and 119 double-annotated total notes. Summary statistics for the corpus are shown in Table 3.3.

Measure	EliIE [67]	Chia [69]	**LCT Corpus**
Disease domain	Alzheimer's Disease	All	**All**
No. of Eligibility Descriptions	230	1,000	**1,006**
No. of Annotations	15,596	68,174	**105,816**
No. of Entity types	8	15	**50**
No. of Relation types	3	12	**51**
Mean Entities per doc.	-	46	**105**
Mean Relations per doc.	-	19	**49**

Table 3.3: **Annotation statistics for EliIE, Chia, and LCT corpora.**

3.3.5 Inter-annotator agreement

Inter-annotator agreement was calculated using F_1 scoring for entities and relations with 20 double-annotated documents. Entity annotations were considered matching only if entity types and token start and end indices matched exactly. Relations annotations were similarly considered matching only if relation type and token start and end indices of both the subject and target matched exactly.

Initial inter-annotator agreement using the 20 training documents was 76.1% for entities and 60.3% for relations. Inter-annotator agreement improved slightly to 78.1% (+2%) for entities and 60.9% (+0.6%) for relations in the 99 additional double-annotated documents, indicating reasonably high annotator agreement considering the complexity of the annotation task.

3.4 Evaluation

To evaluate baseline predictive performance on the LCT corpus, we first created a randomly assigned 80/20 split of the corpus, with 804 documents used for the training set and 202 for the test set. For entity prediction, we trained NER models using biLSTM+CRF and BERT [58] neural architectures. For BERT-based prediction, we used two pretrained models trained on published medical texts, SciBERT [86] and PubMedBERT [87]. We chose these models as both had been shown to perform reasonably well for various tasks related to biomedical NLP. For both biLSTM+CRF and BERT predictions, we trained one model to predict general entities and another for fine-grained entities.

For relation extraction, we evaluated SciBERT for sequence classification as well as a modified BERT architecture, R-BERT, following methods developed by Wu & He [88], also using the pretrained SciBERT model. The R-BERT architecture expects additional tokens in a given text input, [E1] and [E2], to signal the named entities between which to predict a relation. R-BERT modifies the BERT architecture by then deriving a vector representation of each entity using the average of their respective vectors in the BERT final hidden state,

then applies a fully connected layer and softmax layer upon the concatenated entity vectors to predict a relation.

Table 3.4 shows hyperparameters used for each task.

Task	**Architecture**	**Hyperparameter / Embeddings**	**Training Value**
Named Entity Recognition	biLSTM+CRF	Character Dimensions	25
		Token Embedding Dimensions	100
		Learning Rate	0.005
		Dropout	0.5
		Pretrained Embeddings	GloVe [89]
Relation Extraction	BERT & R-BERT	Pretrained Model	SciBert
		Learning Rate	0.00003

Table 3.4: Hyperparameters and pre-trained embeddings used for named entity recognition and relation extraction baseline results. For the NER task, the same architecture and hyperparameters were used for both general and fine-grained entity models. For the relation extraction task, the same hyperparameters were used with both the BERT and R-BERT architectures.

We achieved the highest micro-averaged F_1 score of 81.3% on entities using SciBERT and 85.2% on relations using the R-BERT architecture with SciBERT [1]. Results of representative entities and relations are shown in Tables 3.5 and 3.6.

3.4.1 Annotation quality evaluation

To determine the quality of single-annotated documents compared to those which were double-annotated, we trained NER models (one for general and another for fine-grained

[1] A full listing of baseline prediction results can be found with the annotation guidelines at `https://github.com/uw-bionlp/clinical-trials-gov-annotation/wiki/Named-Entity-Recognition-and-Relation-Extraction-performance`.

Category	Entity	Count	biLSTM+CRF	PubMedBERT	SciBERT
Clinical	Condition	7,087	78.6 / 78.1 / 78.3	76.1 / 79.4 / 77.7	78.4 / 83.3 / 80.8
	Contraindication	142	93.7 / 78.9 / 85.7	77.4 / 80.0 / 78.6	100 / 96.6 / 98.3
	Drug	1,404	76.8 / 81.3 / 79.0	74.1 / 80.9 / 77.4	73.4 / 80.9 / 77.0
	Encounter	302	64.1 / 58.1 / 60.9	51.7 / 61.7 / 56.3	58.3 / 74.4 / 65.4
	Observation	2,558	74.3 / 66.1 / 69.9	67.9 / 73.5 / 70.6	72.1 / 77.6 / 74.7
	Procedure	3,016	68.4 / 75.5 / 71.9	67.0 / 75.9 / 71.2	71.3 / 79.4 / 75.1
Demographic	Age	708	91.3 / 95.4 / 93.3	82.4 / 88.5 / 85.3	99.1 / 98.3 / 98.7
	Birth	27	100 / 80.0 / 88.8	100 / 62.5 / 76.9	100 / 62.5 / 76.9
	Death	35	33.3 / 33.3 / 33.3	0.0 / 0.0 / 0.0	100 / 20.0 / 33.3
	Family-Member	147	40.0 / 19.0 / 25.8	33.3 / 55.5 / 41.6	44.9 / 61.1 / 51.7
	Language	194	92.5 / 96.1 / 94.3	73.8 / 100 / 84.9	96.6 / 93.5 / 95.0
Logical	Negation	952	74.3 / 82.7 / 78.2	60.9 / 73.1 / 66.4	73.5 / 82.9 / 77.9
Qualifier	Assertion	1,157	66.6 / 62.8 / 64.7	56.1 / 58.9 / 57.5	62.1 / 65.8 / 63.9
	Modifier	3,464	65.0 / 58.3 / 61.5	59.2 / 64.0 / 61.5	58.5 / 65.4 / 61.8
	Polarity	360	82.5 / 88.0 / 85.1	74.6 / 67.4 / 70.8	81.4 / 79.5 / 80.4
	Risk	117	93.1 / 96.4 / 94.7	91.3 / 91.3 / 91.3	95.4 / 91.3 / 93.3
	Severity	569	86.8 / 90.8 / 88.7	76.7 / 79.5 / 78.1	86.5 / 94.1 / 90.2
	Stability	397	84.2 / 67.6 / 75.0	79.4 / 75.0 / 77.1	75.3 / 84.7 / 79.7
Comparative	Criteria-Count	33	50.0 / 66.6 / 57.1	28.5 / 40.0 / 33.3	12.5 / 20.0 / 15.5
	Eq-Comparison	5,298	83.1 / 83.8 / 83.4	81.4 / 85.0 / 83.2	85.3 / 89.3 / 87.3
	Eq-Temporal-Period	2,057	88.7 / 89.2 / 88.9	70.0 / 73.9 / 71.9	82.6 / 86.3 / 84.4
	Eq-Temporal-Recency	131	68.7 / 84.6 / 75.8	43.4 / 55.5 / 48.7	50.0 / 66.6 / 57.1
	Eq-Temporal-Unit	1,808	95.1 / 97.6 / 96.4	97.4 / 98.1 / 97.8	98.2 / 99.4 / 98.8
	Eq-Value	3,835	91.8 / 95.3 / 93.5	95.5 / 96.2 / 95.9	96.4 / 97.1 / 96.7
Other	Location	371	68.5 / 58.7 / 63.2	65.4 / 71.6 / 68.3	73.4 / 78.3 / 75.8
-	Total	56,146	80.2 / 79.6 / 79.9	75.3 / 78.7 / 77.0	79.0 / 83.7 / 81.3

Table 3.5: **Baseline entity prediction scores (%, Precision / Recall / F_1).** Corpus-level micro-averaged scores are shown in the bottom row. For brevity a representative sample of entities is shown. *Count* refers to the total count of unique spans annotated in the entire corpus. Entities included in the total count and scores but omitted for brevity are *Acuteness, Allergy, Condition-Type, Code, Coreference, Ethnicity, Eq-Operator, Eq-Unit, Indication, Immunization, Insurance, Life-Stage-And-Gender, Organism, Other, Specimen, Study and Provider.*

Category	Relation	Count	SciBERT	R-BERT+SciBERT
Alternatives and Examples	Abbrev-Of	462	95.2 / 90.9 / 93.0	92.3 / 93.1 / 94.2
	Equivalent-To	516	61.5 / 69.5 / 65.3	59.6 / 67.3 / 63.2
	Example-Of	1,497	94.8 / 92.9 / 93.8	90.5 / 91.7 / 91.1
Clinical	Contraindicates	153	90.9 / 90.9 / 90.9	90.9 / 90.9 / 90.9
	Caused-By	726	63.0 / 86.4 / 72.9	78.6 / 86.4 / 82.3
	Found-By	293	90.4 / 59.3 / 71.7	79.3 / 71.8 / 75.4
	Treatment-For	457	69.2 / 69.2 / 69.2	61.7 / 74.3 / 67.4
	Using	405	73.8 / 83.7 / 78.4	66.6 / 64.8 / 65.7
Logical	And	821	54.1 / 60.0 / 56.9	53.8 / 53.8 / 53.8
	If-Then	261	57.6 / 65.2 / 61.2	55.5 / 65.2 / 60.0
	Negates	984	74.3 / 91.0 / 81.8	74.5 / 88.7 / 81.0
	Or	4,156	85.1 93.2 89.0	88.4 / 92.2 / 90.2
Qualifier	Asserted	1,184	83.7 / 89.0 / 86.3	85.9 / 89.0 / 87.5
	Modifies	3,400	90.9 / 94.2 / 92.5	92.2 / 95.4 / 93.8
	Risk-For	90	92.3 / 85.7 / 88.8	92.8 / 92.8 / 92.8
	Severity	529	80.2 / 96.6 / 87.6	86.3 / 96.6 / 91.2
	Stability	395	76.0 / 92.6 / 83.5	76.4 / 95.1 / 84.7
Temporal and Comparative	After	166	75.0 / 70.5 / 72.7	72.2 / 76.4 / 74.2
	Before	320	70.2 / 86.6 / 77.6	78.1 / 83.3 / 80.6
	Duration	243	59.3 / 79.1 / 67.8	64.5 / 83.3 / 72.7
	During	350	66.6 / 68.7 / 67.6	63.6 / 65.6 / 64.6
	Numeric-Filter	1,957	84.6 / 93.3 / 88.7	85.7 / 92.3 / 88.8
	Minimum-Count	173	64.2 / 69.2 / 66.7	71.4 / 76.9 / 74.0
	Temporality	2,645	80.7 / 90.7 / 85.4	81.8 / 92.2 / 86.7
Other	Location	207	64.2 / 94.7 / 76.6	69.2 / 94.7 / 80.0
-	Total	24,379	80.2 / 88.2 / 84.0	82.5 / 88.0 / 85.2

Table 3.6: **Baseline relation prediction scores (%, Precision / Recall / F_1).** Corpus-level micro-averaged scores are shown in the bottom row. For brevity a representative sample of relations is shown. *Count* refers to the total count annotated in the entire corpus, including relations not shown. The count total excludes general to fine-grained entity relations, which as overlapping spans are not used for relation prediction. Relations included in the total count and scores but omitted for brevity are *Acuteness, Code, Criteria, Except, From, Indication-For, Is-Other, Max-Value, Min-Value, Polarity, Provider, Refers-To, Specimen, Stage, Study-Of and Type.*

entities, as in earlier experiments) using SciBERT with the 887 single-annotated documents and evaluated on the 119 double-annotated documents. The results were a precision of 79.7%, recall of 82.5%, and an F_1 score of 81.4%, which are very close to the highest performance of our randomly split train/test set results shown in Table 3.5. These results indicate relative uniformity and consistency in the corpus across both single- and double-annotated documents.

Training Set	**Test Set**	**Precision**	**Recall**	**F_1**
Manual	Semi-automated	75.4	82.1	78.6
Semi-automated	Manual	80.1	79.9	80.0

Table 3.7: **Results of NER experiments using the manually annotated and semi-automated portions of the corpus.** The manually annotated portion includes 513 documents while the semi-automatically annotated portion is 493 documents.

As the latter near-half (493 documents) of the LCT corpus was automatically annotated, then manually corrected, we also evaluated the quality of the manually annotated portion versus the semi-automatically annotated portion to ensure consistency. We first trained NER models with SciBERT using the manually annotated portion and tested on the semi-automated portion, then reversed the experiment and trained on the semi-automated portion and tested on the manually annotated portion. Results are shown in Table 3.7.

Results of the experiments when training on both the manually and semi-automatically annotated halves of the corpus show comparable results, with the greatest difference being in precision, with the manual annotation-trained model performing slightly worse (-4.7%) in prediction versus the semi-automated annotation-trained model. Overall F_1 scores were similar at 78.6% and 80.0%, suggesting reasonable consistency across the corpus.

3.5 Limitations

The LCT corpus is designed as a granular and robust resource of annotated eligibility criteria to enable models for entity and relation prediction as means of query generation. The corpus does have a number of limitations however which should be recognized. First, the corpus is largely singly annotated, with 119 of 1,006 documents (11%) double annotated and reconciled, while double annotation is generally considered to be the gold standard in the NLP research community. However, the reasonably high F_1 score from experiments to evaluate NER when training on the singly annotated portion of the corpus suggests relative consistency of annotation across both single and double annotated documents. Additionally, entities in roughly half of the LCT corpus (493 documents) were automatically predicted, then manually corrected. This can potentially lead to data bias if predicted entities are not thoroughly reviewed and corrected by human annotators. Similar results from our experiments to detect differences in performance by training on the manually annotated portion versus the semi-automatically annotated portion (F_1 scores of 78.6% and 80.0%) suggest this may not be a significant issue.

While we hypothesize that the demonstrated granularity of the LCT annotation schema facilitates more accurate query generation, this is difficult to measure empirically. One reason for this is that the LCT corpus is designed to enable rule-based query generation, but the accuracy of generated queries depends also on the specific rules used, which may vary among researchers. Further, as the LCT named entities and relations vary from those of other corpora such as Chia, it follows that any rule-based system using LCT models would likely be designed to handle LCT entities and relations specifically. One possible approach would be to have one or more researchers create rule-based systems for query generation based on LCT entities and relations, as well as other corpora, then measure performance based on actual versus expected patients, using a data model such as that of MIMIC-3. This kind of approach however would still be challenging, as rule-development would likely be time-consuming and costly, and researchers participating may not be representative of the general

NLP researcher population.

3.6 Summary

We created the LCT corpus, a gold standard human-annotated corpus of highly granular clinical trial eligibility criteria annotations. The NER and relation extraction models trained on this corpus achieved high performance and enable our subsequent aims.

Chapter 4

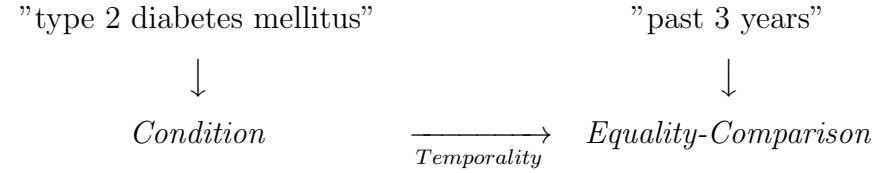
LEAF LOGICAL FORMS CORPUS

4.1 Overview

This chapter examines the motivation for and creation of the LLF corpus, which we train a Seq2Seq model on for transforming input free-text eligibility criteria into logical forms used for generating database queries. Section 4.2 discusses complications in query generation and motivation for logical form structures. Section 4.3 examines related work in eligibility criteria logical representation. In Section 4.4 we describe steps in creating the corpus annotations. Section 4.5 shows results of our experiments, and Section 4.6 provides a summary of this chapter. The resulting Seq2Seq model described here, as well as components described in Chapters 5 and 6 are used within the query system and evaluation described in Chapter 7.

4.2 Motivation

The LCT corpus, as examined in Chapter 3, is a human-annotated corpus of entities and relations within clinical trial eligibility criteria. Entities connected by relations can be imagined as a graph of nodes (entities) and edges (relations) between them. For example, the criterion, *"Diagnosed with type 2 diabetes mellitus within the past 3 years"*, may be represented as a *Condition* node ("type 2 diabetes mellitus") connecting to a *Equality-Comparison* node ("past 3 years") by a *Temporality* edge:



We found SQL query generation using LCT-based graphs to be in many cases error prone and challenging. In particular, nested relations, such as *Or* and *And*, tend to be problematic when represented as edges within a graph. For example, in a criterion such as:

Either of the following:

1. *BMI > 35 in past 6 months*
2. *Previous diagnosis of morbid obesity or obesity with one or more comorbidities*

An *Or* relation would exist between *"BMI > 35"* and *"morbid obesity or obesity with one or more comorbidities"*, as well as a nested *Or* between *"morbid obesity"* and *"obesity"* as well as a nested *And* between *"obesity"* and *"comorbidities"*. In such cases, query generation is challenging as a system must recursively evaluate dependencies between entities to determine the scope of Boolean relations. This problem can be doubly challenging as the complexity of the underlying statement means that predictions of entities and relations by models will also necessarily be imperfect.

An alternative to this approach is to use a so-called intermediate representation (IR), which transforms the original natural language by removing "noise" unnecessary to a given task and which more logically represents underlying semantics, also known as a Semantic Parse [90] (see Herzig *et al* [91] for an examination of IR-based SQL generation approaches). Similar to earlier work using Description Logics, Roberts and Demner-Fushman [92] proposed a representation of questions on EHR databases using a comparatively compact but flexible format using first order logic expressions, for example, representing "Is she wheezing this morning?" as

$$\delta(\lambda x.has\_problem(x, C0043144, status) \wedge time\_within(x, \text{"this morning"}))$$

This style of representation is powerfully generalizable, but also difficult to translate directly into SQL statements as multiple predicates (e.g., *has_problem* and *time_within*) may

correspond to one or many SQL statements, depending on context, complicating direct transformation into queries.

We thus chose a similar intermediate representation (hereafter simply "logical forms") as proposed by Roberts and Demner-Fushman but more closely resembling a nested functional structure in programming languages such as Python or JavaScript and more amenable to SQL generation. A criterion such as "Diabetic women and men over age 65" would be represented by our logical forms as

$$
\begin{aligned}
&intersect(\\
&\quad cond(\text{"}Diabetic\text{"}),\\
&\quad union(female(), male()),\\
&\quad age().num\_filter(eq(op(GT), val(\text{"}65\text{"})))\\
&)
\end{aligned}
$$

An example input criterion and logical form output is shown in Figure 4.1.

4.3 Related Work

A number of logical (sometimes called "formal") representations of eligibility criteria have been explored by researchers. These include ad hoc expression languages such as EON [93], SAGE [94], ERGO [95], and EliXR [65], typically in tree-like dependency structure, Arden Syntax, a prescribed syntax of operators and operations related to health data [96], logic-based languages such as Protégé [97] and SQL, and others [98, 99, 100]. See Weng *et al* [101] for a comprehensive review. As discussed in the previous section, more recent work by Roberts and Demner-Fushman proposed the use of a First Order Logic-based forms [92] which have been demonstrated to work well in tasks such as question answering [102].

4.4 Methods

We developed annotation guidelines for the LLF corpus using a simplification of entities and relations from the preceding LCT corpus[18]. Generally speaking, LCT *entities* correspond

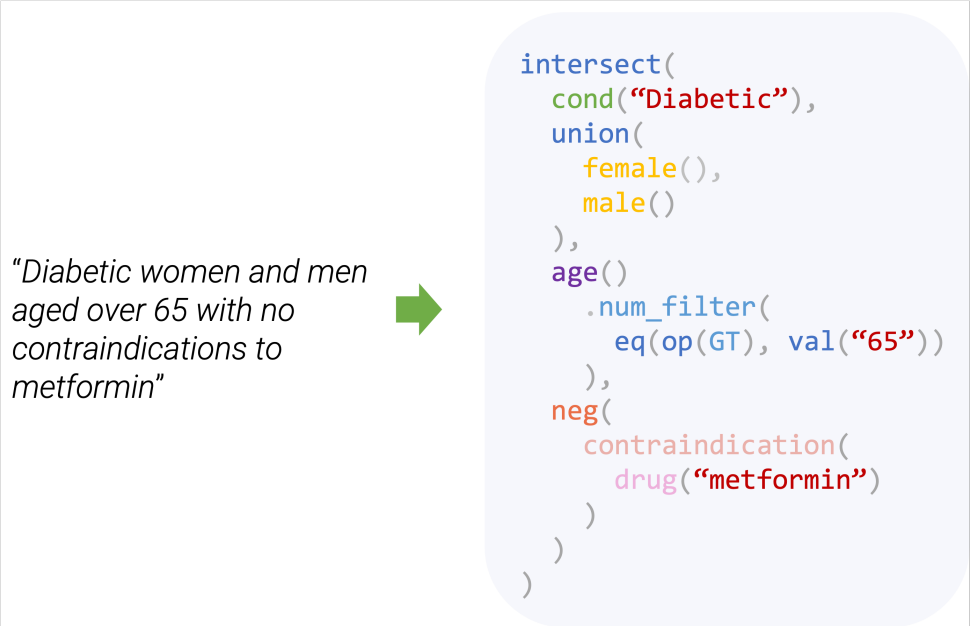


Figure 4.1: An example semantic parse using the LLF corpus annotation schema. The hypothetical input sentence is on the left, while the corresponding logical form is shown on the right. Indentations and colors have been added for readability.

to logical form *functions*, while LCT *relations* correspond to logical form *predicates*. For example, the LCT *Condition* entity has a corresponding *cond()* function, while the *Num-Filter* relation has a corresponding *.num_filter()* predicate. The LLF annotation guidelines can be found at `https://github.com/ndobb/clinical-trials-seq2seq-annotation/wiki`.

We also hypothesized that the performance of predicting logical forms could likely be improved by replacing "raw" tokens in each eligibility criteria with corresponding logical form names derived from named entities from the LCT corpus. For example, given the eligibility criterion:

"*Diabetics who smoke*",

we would replace the named entities for "Diabetics" and "smoke":

cond("Diabetics") who obs("smoke")

using *Condition* and *Observation* annotations in the LCT corpus. We call this substituted text an "augmented" eligibility criteria. The augmented criteria syntax reshapes named entities to more closely resemble expected logical form syntax and allows us to leverage the LCT corpus for logical form transformation.

Creation and annotation of the LLF corpus proceeded in the following steps:

1. We randomly chose 2,000 lines of eligibility criteria from the LCT corpus, limited to only criteria which included at least one named entity and which were not annotated as hypothetical criteria. 30% of the 2,000 lines (600) were randomly chosen among lines with particularly complex entity and relation types, such as *If-Then*, *Before-After*, *Contraindication*, etc.

2. Each annotation file consists of the text "EXC" if exclusion or "INC" if inclusion (line 1), an original "raw" eligibility criteria (line 3), an augmented eligibility criteria (line 5), and an (initially blank) expected logical form equivalent to annotate (line 7). An example annotation is shown in Figure 4.2.

3. 3 informatics graduate students met weekly for 2 months to review annotations. Annotators were initially trained on 20 triple-annotated training annotations.

4. After training, each annotator was assigned a batch of 100 sentences (one per file) and tasked with writing a logical form version of each.

5. After each batch was completed, we executed a quality control script to parse each logical form annotation to ensure consistency. Any syntax errors were reported to and corrected by the annotators.

6. Annotators received additional batches of files to annotate until all 2,000 single-annotated annotations had been completed.

```
'INC'

'-  women age 20 - 34 years ;'

'-  female() age() eq(val("20"), op(BETWEEN), val("34"), temporal_unit(YEAR)) ;'

intersect(
    female(),
    age()
        .num_filter(
            eq(val("20"), op(BETWEEN), val("34"), temporal_unit(YEAR))
        )
)
```

Figure 4.2: An example LLF corpus annotation. The annotation file is saved in JavaScript (.js) format, which enables syntax highlighting and validation to assist annotators. Whether a given criterion was an inclusion or exclusion criteria is indicated at the top, followed by the original raw text, then augmented text. The final annotated logical forms are shown last.

The pair-wise mean inter-annotator agreement by BLEU score was 82.4%. After annotations were completed, we experimented with predicting logical forms by fine-tuning T5 [103] Seq2Seq models. The T5 architecture and pre-trained models are widely used for and achieve at or near state-of-the-art for many machine translation and semantic parsing tasks.

Following earlier work on task-oriented dialog semantic parsing structures in the domain of digital assistants, we also experimented using various alternative input-output syntax styles from our original logical forms:

1. **Shift-Reduce**. Einolghozatic *et al.* [104] used square brackets instead of parentheses and blank spaces instead of commas. We followed Rongali *et al's* suggestion to add a trailing repeat of function names to improve performance.

2. **Pointer**. Rongali *et al.* [105] found that replacing input tokens with "@ptr_{index}", where *index* corresponds to a token's sequential position in the input text improved performance in their semantic parsing task. We modified this approach by omitting the characters "ptr" and using the sequential position of the quoted span as our index rather than individual token positions.

4.5 Results

We used a randomly sorted 70/20/10 train/test/validation split of the LFF corpus to fine-tune the pretrained $T5_{base}$ model using combinations of these syntax styles. We call our gold standard annotated logical form syntax and augmented text inputs "Standard" style. Example inputs, outputs, and training results are shown in Table 4.1.

We found that our Standard logical forms achieved the highest performance using both BLEU [106] and ROUGE-L [107] scores, two commonly used metrics in measuring Seq2Seq performance. As can be seen in comparing rows 1 and 2 on Table 4.1, using our "augmented text" as input (i.e., replacing raw tokens with function names corresponding to named entities, row two) significantly improved performance, with a BLEU score +14.7% higher as compared to raw text as input. As it was the highest-performing syntax style and also the most straightforward to parse, we chose to use the Standard logical form style as our IR for this project.

4.6 Summary

We created the LLF corpus, a gold standard human-annotated corpus of 2,000 eligibility criteria and corresponding logical forms. The LLF syntax was developed based on study of past work and entities and relations present in the LCT corpus. Our best performing prediction method using a fine-tuned T5 model achieved > 93% BLEU score.

Syntax Style	**Example Input**	**Example Logical Form**	**BLEU**	**ROUGE-L**
Raw-text→ Standard	Diabetics who smoke	*intersect*(*cond*("*Diabetics*"), *obs*("*smoke*"))	78.7	79.1
Standard	cond("Diabetics") who obs("smoke")	*intersect*(*cond*("*Diabetics*"), *obs*("*smoke*"))	**93.5**	**92.3**
Standard+ Pointer	cond(@1) who obs(@2)	*intersect*(*cond*(@1), *obs*(@2))	93.3	91.2
Shift-Reduce	[cond "Diabetics" cond] who [obs "smoke" obs]	[*intersect* [*cond* "*Diabetics*" *cond*] [*obs* "*smoke*" *obs*] *intersect*]	89.8	91.7
Shift-Reduce+ Pointer	[cond @1 cond] who [obs @2 obs]	[*intersect* [*cond* @1 *cond*] [*obs* @2 *obs*] *intersect*]	89.4	90.4

Table 4.1: Example inputs and logical form syntax styles with fine-tuning performance results using the $T5_{base}$ model.

Chapter 5
KNOWLEDGE BASE

5.1 Overview

Storing biomedical information such as conceptual mappings, controlled vocabularies and terminologies, synonyms, hyponyms, and hypernyms enables a great variety of useful capabilities for a natural language interface for cohort discovery. For example, enabling users to simply specify "Patients without contraindications to Metformin"—without needing to exhaustively list what such contraindications may be—saves user time and energy, and also possibly includes criteria they may not be aware of.

This chapter describes the development of our KB, which combines a variety of sources using a graph database of Resource Description Framework (RDF) [108] triples. Section 5.2 describes our motivation for creating this resources, while Section 5.3 describes the data sources used and methods for KB population. Section 5.4 summarizes the work described in this chapter.

5.2 Motivation

While non-specific eligibility criteria can be categorized in a variety of forms, we found a number of frequently used patterns during the development of the LCT and LLF corpora. We focused on enabling reasoning upon the following non-specific criteria categories:

1. **Treatments for Conditions**, such as *"treated for myocardial infarction"*. Determining whether a patient was treated for this requires first determining what possible treatment options exist for said condition.

2. **Contraindications to Treatments**, for example *"Contraindicated for MRI"* or *"With*

known contraindications to ACE inhibitors". Contraindicated concepts must therefore be searched for procedures, surgical treatments, and drugs (including possible drug-drug interactions).

3. **Observations indicating a Risk**, such as *"At risk for suicide"* or *"At risk for heart attack"*. Possible observations indicating risk thus include those for self-harm (such as depression) as well as underlying disease.

4. **Signs and Symptoms of a Condition**. These may include criteria such as *"Symptoms of depression"* or *"Showing signs of COVID-19"*.

5. **Conditions affecting a Physiological Function**, for example *"Conditions affecting respiration"*.

6. **Indicated Treatments for a Condition**, such as *"Indicated for anticoagulation therapy"* or *"Indicated for endoscopical drainage"*.

Beyond facilitating reasoning on non-specific criteria, KBs also serve a vital role in other useful functions, such as determining relevant ICD-10 or SNOMED codes for a given diagnosis, or traversing a concept heirarchy to find relevant related concepts (e.g., determining a range of possible conditions associated with primary hypertension).

5.3 Methods

5.3.1 Data Sources

Given its widespread use and breadth of vocabulary and terminology coverage, we chose to use the UMLS as the core of our KB. Despite the broad coverage of biomedical concepts it represents, however, the UMLS lacks certain data elements, specifically relations, needed to enable our goals for non-specific concept reasoning. For example, the UMLS contains only limited data related to contraindications and symptoms. We thus evaluated informatics

literature to find publicly available datasets and ontologies which may help fill these gaps. Importantly, as within the UMLS a Concept Identifier, or "CUI", forms the underlying global identifier linking the various biomedical source systems, we focused on the integration of sources directly (via CUI mappings) or indirectly (via ICD-10 or other coding systems) linkable to the UMLS. We integrated the following ontologies and data sources:

1. **The Disease Ontology (DO)** [20]. The DO is a comprehensive KB of human diseases. The DO features cross references to ICD-9, ICD-10, SNOMED, and other systems. For the purposes of this project, we leverage the DO's listing of symptoms for diseases with mappings to the Symptom Ontology (discussed next). For example, DOID_0080642 ("Middle East respiratory syndrome") includes SYMP_0000242 ("cough with bloody sputum") as a symptom. Symptoms related to COVID-19 were added to the DO at our request [1].

2. **The Symptom Ontology (SO)** [21]. The SO is a large KB of symptoms. Like the DO, it includes external references to other vocabularies, including the UMLS. We leverage the DO and SO jointly by mapping both to UMLS CUIs.

3. **The COVID-19 Ontology** [22]. The COVID-19 ontology includes a variety of data elements and mappings related to COVID-19, such as transmission vectors, signs and symptoms, diagnostic methods, prevention and control, as well as genetic and molecular processes. For our purposes, we leverage risk factors for COVID-19 (COVID_0000207), which in turn link to DO identifiers.

4. **Potential Drug-Drug Interactions (PDDI)** [23]. The PDDI is a large data set of potential drug-drug interactions. The PDDI was developed by merging 14 different sources, including from clinical information sources, NLP corpora and pharmacovigilance sources. We used the PDDI for reasoning upon contraindications to medications.

[1] https://github.com/DiseaseOntology/HumanDiseaseOntology/issues/916

5. **The Disease-Symptom Knowledge Base (DSKB)** [25]. The DSKB is a listing of diseases and corresponding symptoms derived using the NLP system MedLEE [109] on 25,074 discharge summaries from New York-Presbyterian Hospital. A random subset of the NLP-derived information was reviewed by a clinician for an overall recall of 90% and precision of 92%. The DSKB contains UMLS CUIs and was thus directly linkable to our KB.

6. **LOINC2HPO** [24]. LOINC2HPO is a data set of approximately 3,000 laboratory tests with corresponding phenotypes depending on their results (e.g., high, low, normal, positive, negative) linked to the Human Phenotype Ontology (which in turn is included in the UMLS). While not directly related to our reasoning use cases, we incorporated LOINC2HPO in order to find patients who may not have diagnosis codes for a given condition but whom a condition could be inferred via laboratory test results.

We further integrated a number of vocabulary mappings which we found potentially useful and not present within the UMLS. For example, an eligibility criteria may specify "Patients mechanically ventilated", which would be normalized to concept C0199470 ("Mechanical Ventilation") using MetaMapLite (discussed in Chapter 7). Within the UMLS however, C0199470 is not associated with any ICD-10 PCS codes, while within EHRs, mechanical ventilation events are recorded using ICD-10 PCS codes such as "5A19352", which relate to a different concept, C2695822 ("Respiratory Ventilation, Less than 24 Consecutive Hours"). As these concepts and codes are not linked within the UMLS, our KB would not be able to appropriately generate a database query to find mechanically ventilated patients. Using the following sources, however, enables our KB to fill this gap (among many others):

1. **ICD-9-CM to and from ICD-10-CM and ICD-10-PCS Crosswalk or General Equivalence Mappings** [26]

2. **ICD-9-CM Procedure Codes to SNOMED CT Map** [28]

3. **SNOMED CT to ICD-10-CM Map** [29]

4. **ICD-9-CM Diagnostic Codes to SNOMED CT Map** [27]

The integration of these additional mappings enable our KB to derive codes for a wide variety of use cases.

5.3.2 Graph Database Population

A KB is an abstract, rather than technical concept, which may be instantiated as a relational or document database, graph, or other data structure. The practical needs and use cases related to eligibility criteria in this project drew us toward the use of a graph database representation. An extensive comparison of graph databases versus relational or other databases is outside the scope of this discussion. In brief however, both graph and relational databases are capable of storing data encompassing the same linkages and tuples [110], though by different means. Generally speaking, graph databases often enable a more succinct query syntax (i.e., the same data extract can be accomplished using graph database queries linking tuples than one using a large number of SQL JOIN operations) and further do not necessitate a predefined schema, making them ideal for linking and exploring heterogeneous data sources.

We used GraphDB [111], a robust commercial graph database platform which is free for research use as our KB. To populate our KB, we modified open-source extraction scripts used for BioPortal [74] which extract UMLS data representations from a SQL database and output into the RDF-compatible format, Terse RDF Triple Language (Turtle) [112]. In the original BioPortal extraction scripts[2], UMLS CUIs are represented as strings, such as "C0150840", rather than Internationalized Resource Identifiers (IRIs), such as

$$< http://bioportal.bioontology.org/ontologies/umls\#C0150840 >.$$

IRIs serve as critical structures in RDF for linking data sets together. Storing CUIs as strings results in duplication of data, performance degradation (graph databases optimize retrieval

[2]https://github.com/ncbo/umls2rdf

speed by caching IRI-based tuple linkages), and more complicated query syntax which runs somewhat contrary to RDF specifications [108]. We therefore modified[3] the BioPortal scripts to treat CUIs as IRIs. We similarly converted data sources not stored in an RDF-related format (such as CSV, HTML, or SQL) to Turtle files by Python script. All Turtle files were subsequently loaded into GraphDB.

5.3.3 SPARQL Query Development

After data were loaded in the KB, we developed SPARQL queries for each reasoning and hierarchy traversal use case. SPARQL is a graph query language developed by the W3C and features a syntax relatively similar to SQL [113]. For SPARQL queries related to reasoning use cases, development generally involved exploration of UMLS relations across various vocabularies, in addition to non-UMLS sources in our KB. In cases where more than one relation type was found, we traversed tuples from multiple vocabularies using $UNION$ or $IN(< relation1, relation2, ... >)$ syntax.

Below is an example of a SPARQL query for extracting concepts related to a sign or symptom. The script accepts a @*cui* parameter and traverses the KB, outputting sign and symptom-related CUIs and metadata from the DSKB, DO, and Omaha System ontologies:

```
# get_signs_and_symptoms_by_condition_cui.sparql
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX skos: <http://www.w3.org/2004/02/skos/core#>
PREFIX umls: <http://bioportal.bioontology.org/ontologies/umls#>
PREFIX oms: <http://purl.bioontology.org/ontology/OMS#>
PREFIX dis: <http://purl.obolibrary.org/obo/DOID_>
PREFIX dskb: <http://purl.bioontology.org/ontology/DSKB#>

SELECT ?cui ?label ?sab ?code ?code_label ?tui
WHERE
{
    ?subj_iri skos:notation ?subj_cui .
    ?subj_iri skos:prefLabel ?subj_label .
```

[3] https://github.com/ndobb/umls2rdf

```
    ?subj_iri ?rel1       ?obj1_iri  .
    ?obj1_iri skos:notation ?cui     .
    ?obj1_iri skos:prefLabel ?label  .
    ?obj1_iri umls:code   ?code_iri  .
    OPTIONAL
    {
        ?code_iri skos:notation ?code    .
        ?code_iri skos:prefLabel ?code_label .
        ?code_iri rdfs:domain  ?ont_iri   .
        ?ont_iri rdfs:label   ?sab       .
        ?obj1_iri umls:sty    ?sty_iri   .
        ?sty_iri skos:notation ?tui      .
    }
    FILTER (?subj_cui = @cui)
    FILTER (?rel1 IN (
               dskb:hasSymptom,
               dis:hasSymptom
               oms:has_sign_or_symptom
          ))
}
```

5.4 Limitations

Our KB has a number limitations. First, the KB does not have a means of representing severity, probability, significance, or other nuanced relationships between a subject concept and reasoned concepts. For example, the PDDI data source lists certain drug-drug interactions as low in significance or only possible, and thus including those as contraindications may be unhelpful in certain cases. Chapter 8 demonstrates that users have the ability to edit reasoned concepts in our web application, which we believe makes this an acceptable trade-off. Second, our KB is capable of providing only lists of UMLS concepts as output, which in certain cases is not ideal. For example, our KB may return a concept such as "Old age" (C1999167) or "Low Platelet Count" (C5201036), which would be translated downstream into SNOMED or other codes for a structured query. We suspect that in most cases though, determining that a patient is over a certain age or has low platelet counts would be more accurately performed by querying structured demographic or laboratory result data directly (e.g., *"WHERE patient_age* $>= 65$").

In a related fashion, more complex reasoning tasks, such as for patients "Indicated for

bariatric surgery", require complex Boolean logic to be represented appropriately. For example, the Mayo Clinic describes indications as follows, using multiple levels of nested Boolean logic:

1. *Body mass index (BMI) is 40 or higher, called extreme obesity, OR*
2. *BMI is 35 to 39.9, called obesity, and have a serious weight-related health problem, such as type 2 diabetes, high blood pressure or severe sleep apnea.*
3. *In some cases, BMI is 30 to 34 and have serious weight-related health problems.*[4]

Our KB is unable to represent reasoning logic like this, as it can respond with only a list of UMLS concepts. In future work, we may examine adaptation of our KB to respond with logical forms (Chapter 4), which would be capable of representing these more complex cases.

5.5 Summary

This chapter describes the motivation and development of our KB, instantiated as a graph database which incorporates the UMLS and numerous other linked sources. Our KB forms the underlying foundation which enables reasoning on non-specific criteria (described in Chapter 7).

[4]Adapted from `https://www.mayoclinic.org/tests-procedures/bariatric-surgery/about/pac-20394258`

Chapter 6

SEMANTIC METADATA MAPPING

6.1 Overview

Any system which aims to generate queries to fulfill an aim or process will be necessarily limited in utility if it operates with only a single static database schema. Within the health domain, data structures and vocabularies are increasingly complex and diverse, and systems capable of generalizing to other data models and institutions stand to be more impactful. Even the OMOP common data model, though widely used, is often updated to include custom database tables and data to fulfill various project-specific needs [114, 115, 116, 117, 118, 119, 120].

This chapter explores what we call a Semantic Metadata Mapping, or SMM, which we leverage as a data-model agnostic means of generating queries for eligibility criteria. Section 6.2 explores related work. Section 6.3 discusses SMM structure and configuration. Section 6.4 summarizes the work of this chapter.

6.2 Related Work

Many recent efforts at creating machine-readable data and resources for cross-system communication originate with the Semantic Web [121], a set of standards undergirded by ontologies, taxonomies, and inference rules designed for the Internet. As explained by the creators of the Semantic Web:

> *"An ontology may express the rule "If a city code is associated with a state code, and an address uses that city code, then that address has the associated state code." A [Semantic Web] program could then readily deduce, for instance, that a*

Cornell University address, being in Ithaca, must be in New York State, which is in the U.S., and therefore should be formatted to U.S. standards." [121]

The logic behind data model-agnostic query generation using an SMM is similar. For example, suppose an eligibility criteria specifies that patients identify as Hispanic. If an SMM contains a database column, $IsHispanic$, where a given tuple of value "1" means the tuple represents the UMLS concept of Hispanic (C0086409), a program can deduce then that a SQL query with a WHERE clause of *"WHERE IsHispanic = 1"* should retrieve patients who identify as Hispanic. This relatively simple logic is inspired by concepts of the Semantic Web and forms the basis of SMMs. Where SMMs differ most clearly, however, is the use of the UMLS as a means of managing metadata. To the best of our knowledge, this is the first project to do so to enable database query generation.

Using metadata for the purpose of generating database queries has been explored in previous research. Bizer and Seaborne [122] demonstrated that a simple ontology of database schema elements could be used to automatically generate SQL queries. Sequeda *et al* proposed similar methods using First Order Logic and and created a publicly available rule engine for the research community [123, 124]. Knoblock *et al* and Gupta *et al* demonstrated the potential of such method for integrating heterogeneously structured databases of protein, gene, and metabolic pathway data [125, 126].

6.3 Methods

An SMM includes a listing of available databases, tables, columns, and so on within a given database schema. These database artifacts are "tagged" using a subset of UMLS concepts, many from the National Cancer Institute (NCI) and Health Level 7 (HL7) vocabularies. Examples of categories and concepts possible within an SMM include:

- **Demographic** - Age (C0001779), Ethnic Group (C00015031), Language (C0023008), Female (C0086287), Male (C0086582), Transgender (C3266856), Hispanic (C0086409).

- **Identifiers** - Patient (C5236161), Encounter (C3865224)
- **Metadata** - Concept Mapping (C3858752), Code (C0805701), Code System (C2347818), Quantitative Value (C0392762).
- **Polarity** - Normal (C0205307), Abnormal (C0205161), Positive (C1514241), Negative (C0205160), Low (C0205251), High (C0205250).
- **Vital Status** - Living (C4551704), Deceased (C0011065).

Alternatively, a column may also be tagged using a UMLS Source Abbreviation, or SAB, such as ICD-10, SNOMED, LOINC, or RXNORM. Tagging a column as an SAB indicates that the tuple contents of a column represent codes associated with a given source system. For example, a column which contains diagnosis codes such as E11.62, I25.10, and others (ICD-10 codes) would have an SAB tag of "ICD10CM". SABs supported by our SMMs are:

- **Current Procedural Terminology (CPT)**
- **Healthcare Common Procedure Coding System (HCPCS)**
- **ICD-9**
- **ICD-10**
- **ICD-10 Procedure Coding System (ICD-10 PCS)**
- **LOINC**
- **National Drug File (NDC)**
- **NCI**
- **RxNorm**

- **SNOMED**

A column tagged by a UMLS concept or SAB can also be conditional, or in other words, only true given some other logic also being true. For example, Table 6.1 shows a hypothetical database table of diagnosis codes. Column *code* contains codes from various vocabularies, including ICD-9, ICD-10, and SNOMED.

patient_id	**coding_system**	**code**	**dx_name**
1	ICD-10	I50	Heart Failure
2	ICD-10	E87.70	Fluid Overload
3	SNOMED	4270721002	AIDS Associated Disorder
4	ICD-9	493.0	Asthma

Table 6.1: Hypothetical database table for patient diagnoses. Codes from a variety of different vocabularies are shown under the *coding_system* column.

In this case, *code* would be tagged with 3 conditional SABs, ICD-9, ICD-10, and SNOMED, with each SAB tag's condition defined as a SQL WHERE clause in the form of a string. For example, the "ICD10CM" tag would be paired with the condition, "WHERE coding_system = 'ICD-10'". This is a machine-readable equivalent of, "The *code* column represents an ICD-10 code only if the *coding_system* column in the same record is of value 'ICD-10'". Table 6.2 shows a more complete configuration example of all 3 SABs for the column.

ColumnName	**SAB**	**SqlWhere**
code	ICD9CM	coding_system = 'ICD-9'
code	ICD10CM	coding_system = 'ICD-10'
code	SNOMED	coding_system = 'SNOMED'

Table 6.2: Example SMM configuration for the *code* column.

Figure 6.1: Two hypothetical database schema to generate queries for platelet counts (shown in logical form after normalization). This example illustrates the flexibility of our SMM system (represented here in JSON format) in adapting to virtually any data model. On the left, "Tall Table Structure", platelet counts must be filtered from within a general purpose "labs" table. Our KB recognizes that labs may be stored as LOINC codes and the corresponding SMM indicates that records in this table can be filtered to LOINC values. On the right, "Pivoted Table Structure", platelet counts are stored as a specific column in a "complete_blood_counts" table, and thus can be directly queried without further filtering. Additional metadata, columns, tables, types and so on needed in SMMs are omitted for brevity.

The use of UMLS concepts, SABs, and optional conditions for each enables query generation across a wide variety of potential data models and database schema. An example of this can be seen in Figure 6.1, which shows strategies by which a given criterion can be used to generate schema-specific queries by leveraging different SMMs. In cases where there is more

than one means of querying a given concept (e.g., two SQL tables for diagnosis codes), the SMM returns both tables, which may be combined in a UNION statement downstream. The query generation algorithm for SMMs is shown in Algorithm 1. SQL queries for the "Tall Table Structure" (left example) in Figure 6.1 would be generated using the *directSqlSets* array in Algorithm 1, while the "Pivoted Table Structure" (right example) would be generated using *encodedSqlSets*. As the algorithm checks for mappings in both cases at runtime, the algorithm does not need to be recompiled to handle different database structures.

6.4 Limitations

Our SMM implementation is limited in a number of ways, most notably in that our tagging schema is not entirely machine readable. Specifically, the use of optional conditions in the form of SQL WHERE clauses implemented as simple strings (e.g., "WHERE IsPrimaryDx = 1") means that an SMM does not carry granular semantic information of actual filtering logic or tuple-level values, but rather simply trusts that a given SQL condition will filter as expected. A more robust future SMM implementation may annotate tuple-level values semantically at a more granular level, though this would necessarily entail far more work on the part of the annotator in exhaustively describing every possible value of every relevant column in a database.

6.5 Summary

This chapter examined our system for data model-agnostic query generation using SMMs, an encoding system where metadata on various clinical database schema elements are tagged using UMLS concepts and SABs.

Algorithm 1: Algorithm for generating SQL queries by mapping concepts for a given logical form and SMM. Actual SQL syntax generation steps and certain steps are omitted for brevity. Each concept within a logical form is evaluated for direct mappings to SMM columns within a database and encodable mappings to SABs. *SMM.columns* and *SMM.sabs* are dictionaries of UMLS concepts which return zero or one database columns. The SQL Set objects in *directSqlSets* and *encodedSqlSets* are not strings but rather strongly typed objects representing a single SQL statement. Actual SQL compilation is performed in a subsequent process.

```
1  directSqlSets ← []  /* SQL from DB columns mapped to specific CUIs      */
2  encodedSqlSets ← []  /* SQL from DB columns which encode SABs          */
3  for concept in LogicalForm.concepts do
4      if concept in SMM.columns then
5          column ← SMM.columns[concept]
6          sql ← GenerateConceptSql(concept, column)
7          directSqlSets.push(sql)
8      end
9      for sab in SMM.sabs do
10         if concept encodable as sab then
11             column ← SMM.sabs[concept]
12             sql ← GenerateEncodedSabSql(concept, column)
13             encodedSqlSets.push(sql)
14         end
15     end
16 end
17 return directSqlSets ∪ encodedSqlSets
```

Chapter 7
QUERY GENERATION

7.1 Overview

This chapter describes our overall system for automatic query generation for clinical trial eligibility criteria using a natural language interface, which we call LeafAI. This system incorporates NER models (trained on the LCT corpus, described in Chapter 3), a Seq2Seq model (trained on the LLF corpus, described in Chapter 4), a KB (Chapter 5), and methods for database schema metadata tagging (Chapter 6). Section 7.2 recapitulates the motivation for our system design. Section 7.3 summarizes relevant related research described earlier in Chapter 2. Section 7.4 describes methods for query generation leveraging components discussed in preceding chapters. Experimental setup and evaluation methods are described in Section 7.5, with results shown in Section 7.6. A discussion of results, implications, and limitations is presented in Section 7.7. Section 7.8 summarizes the content of this chapter.

7.2 Motivation

Identifying groups of patients meeting a given set of eligibility criteria is a critical step for recruitment into RCTs. Often, clinical studies fall short of recruitment goals, leading to time and cost overruns or challenges in ensuring adequate statistical power [34, 36]. Failure to recruit research subjects may result from a variety of factors, but often stems from difficulties in translating complex eligibility criteria into effective queries that can sift through data in the EHR [15]. Despite these difficulties, RCT investigators increasingly rely on EHR data queries to identify research candidates instead of labor-intensive manual chart or case report form review [127]. At the same time, the amount and variety of data contained in EHRs is increasing dramatically, creating both challenges and opportunities for patient recruitment

[128]. While more granular and potentially useful data are captured and stored in EHRs now than in the past, the process of accessing and leveraging that data requires technical expertise and extensive knowledge of biomedical terminologies and data models.

Given these challenges, automated or semi-automated means of identifying eligible patients in reducing time and costs in human labor while leveraging more complex but useful data are appealing. In particular, NLP-based cohort discovery methods could be especially valuable since they can key on eligibility criteria described in natural language, a medium that clinicians, researchers and investigators already use.

7.3 Related Work

In this section we briefly summarize recent research in this domain, described earlier in Chapter 2.

Much research has explored methods for query generation, including encoder-decoder neural architectures for transforming clinical natural language questions into SQL queries [53, 54, 55, 56, 12]. The most prominent recent system and most comparable to our approach is Criteria2Query [6, 8]. Criteria2Query utilizes a combination of rule-based and neural modules to generate queries: a CRF-based NER model trained on a corpora of 230 Alzheimer's Disease eligibility criteria documents [67], relation extraction using Dijkstra's algorithm [129], BERT [58] for negation detection, Boolean logic detection using heuristics and dependency parsing, a Lucene-based [59] OMOP mapping tool, Usagi [130] for named entity normalization, and SuTime [83] for temporal expression normalization. For query generation, Criteria2Query composes the resulting query representation into JSON and leverages the Observational Health Data Sciences and Informatics (OHDSI) ATLAS [131] API to generate a SQL query.

7.3.1 Gaps and opportunities

Most programs capable of generating database queries do so for only a single database schema, such as OMOP or MIMIC-III [57]. This lack of flexibility limits their capability to accommodate real-world project needs [114, 115, 116, 117, 118, 119, 120], such as adding new

database tables to OMOP for cancer staging [114]. Moreover, most methods, particularly those using direct text-to-SQL deep learning approaches, tend to generate relatively simple SQL statements with few JOINs or nested sub-queries and typically no support for UNION operators and so on. This relative simplicity contrasts with the complexity of real-world EHR databases, which may contain dozens or even hundreds of tables using various vocabularies and mappings. Furthermore, direct text-to-SQL methods are bound to SQL syntax, and thus incapable of querying other systems such as Fast Healthcare Interoperability Resources (FHIR) [132]. Additionally, few of the methods described provide support for complex logic such as nested Boolean statements or temporal sequences, and none support reasoning on non-specific criteria (e.g., "diseases that affect respiratory function"), phenomena common to study eligibility criteria [15, 16]. Perhaps most importantly, to the best of our knowledge, only one previous work has been tested in terms of matching patients actually enrolled in clinical trials [9], and none have been directly compared to the capabilities of a human database programmer.

7.4 Methods

7.4.1 System Architecture

The LeafAI query engine was designed using a modular, micro service-based architecture with a central Application Program Interface (API) which orchestrates end-to-end query generation. Inter-module communication is performed using gRPC [133], a robust open-source remote procedure call framework which enables language-agnostic service integration. This allows individual modules to be implemented (and substituted) in programming languages and using libraries well-suited to a given task. We deploy each module as a Docker [134] container. A diagram of the LeafAI query engine architecture is shown in Figure 7.1.

At a high level, query generation is performed in the following steps:

1. A query request is received by the API in the form of inclusion and exclusion criteria as free-text strings.

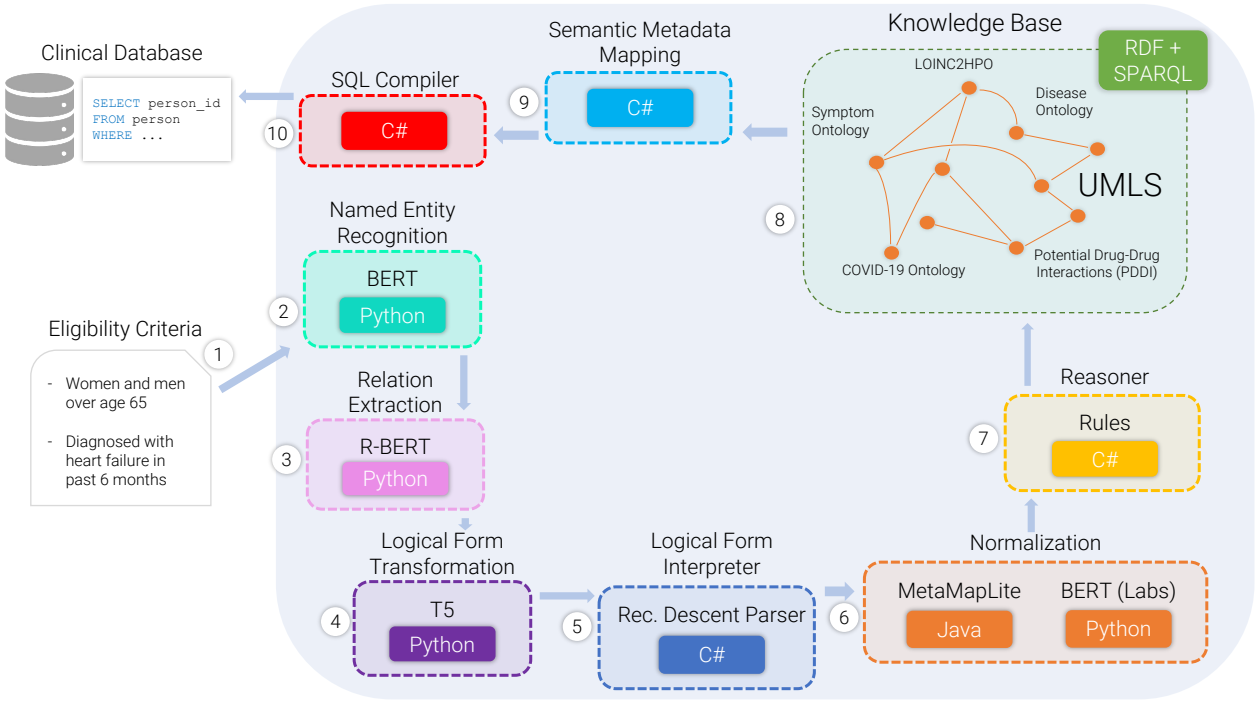


Figure 7.1: LeafAI query architecture. Inter-module communication is performed using the gRPC framework. Individual modules are deployed as Docker containers and communicate solely with the central API, which orchestrates query generation and handles query generation requests.

2. The input texts are tokenized and named entity recognition is performed to determine spans of text representing conditions, procedures, and so on.

3. Relation extraction is performed to determine relations between the entities, such as *Caused-By* or *Numeric-Filter*.

4. The input texts are transformed into augment text by replacing spans of "raw" text with logical form names. For example, "Diagnosed with diabetes" would become "Diagnosed with cond("diabetes")." The resulting input texts are in turn transformed into an output logical representation using a Sequence to Sequence (Seq2Seq) architecture,

in the form of a string.

5. A logical form interpreter module implemented as a recursive descent parser [135] reads the logical form string input and instantiates it as an abstract syntax tree (AST) of nested in-memory logical form objects.

6. "Named" logical form objects (i.e., specified with quoted text, such as "cond("diabetes")") are normalized into one or more corresponding UMLS concepts. UMLS child concepts are also added using our KB. For example, "cond("type 2 diabetes")" would also include concepts for type 2 diabetes with kidney complications (C2874072).

7. Working recursively inside-to-outside the AST structure, each logical form object calls a *Reason()* method which executes various rules depending on context.

8. Each reasoning rule is performed as one or more pre-defined SPARQL queries to the KB, concept by concept.

9. The final normalized, reasoned, logical form AST is thus a nested structure of UMLS concepts. Each AST criterion is mapped to zero or more corresponding entries in the semantic metadata mapping (SMM), which in turn lists meanings, roles, and relations of a database schema in the form of UMLS concepts.

10. The final mapped AST object is transformed into a series of database queries, one per line of eligibility criteria text. The output SQL query can either be executed directly on a database or returned to the API caller.

Figure 7.2 illustrates an example of this process. In the following subsections we examine these steps in detail.

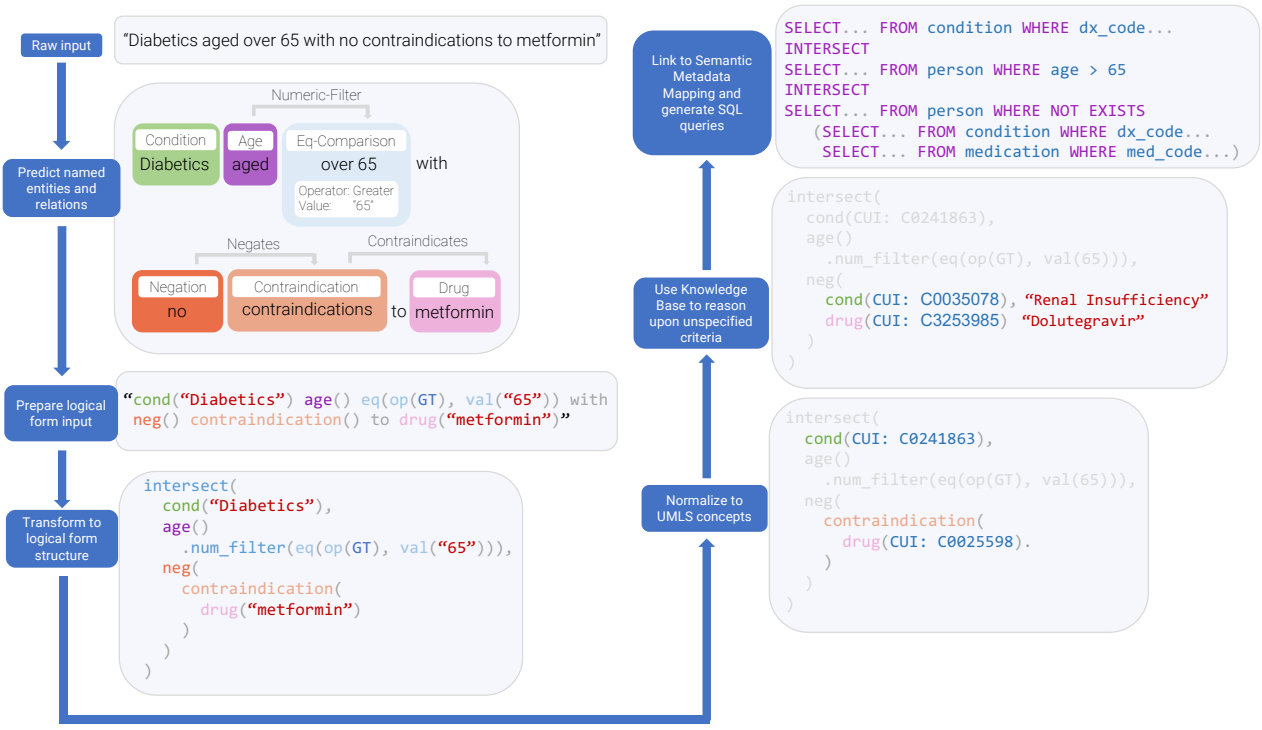


Figure 7.2: LeafAI query generation processes

7.4.2 Named entity recognition and relation extraction

We used the two best-performing BERT-based NER extractors from Chapter 3, one each for LCT general- and fine-grained-entities. Next, we perform relation extraction between named entity pairs similarly using a BERT-based model also trained on the LCT corpus.

7.4.3 Logical form transformation

As discussed in Chapter 4, one of the core challenges of generating queries for eligibility criteria is the problem of logical representation. Generating queries directly based on named entities and relations alone, while practical, performs poorly in cases of nested or particularly complex logic.

After NER and relation extraction are performed, we leverage our best-performing T5

Seq2Seq model fine-tuned for predicting logical forms on the LLF corpus. As inputs to the Seq2Seq model we use the original eligibility criteria with named entity spans replaced by logical form representations, as we found this improved performance compared to training with raw inputs (described in Chapter 4). For example, the criterion

"Women 55 and older diagnosed with Hepatitis C"

would be transformed to the augmented criterion

"female() eq(op(GTEQ), val("55")) diagnosed with cond("Hepatitis C")"

and finally returned as the logical form

$$
\begin{aligned}
&intersect(\\
&\quad female(), \\
&\quad age().num\_filter(eq(op(GTEQ), val("55"))) \\
&\quad cond("Hepatitis\ C"), \\
&)
\end{aligned}
$$

from our Seq2Seq model. The returned logical form string would then be instantiated into an abstract syntax tree (AST) of nested in-memory logical form objects using a recursive descent parser [135] within our API.

7.4.4 Concept normalization

We normalize "named" logical forms to UMLS concepts using MetaMapLite [136, 137], a widely used [138, 139, 140, 141, 142] and reasonably well performing rule-based application for normalization of clinical texts. We consider a logical form "named" if it contains a free-text value surrounded by quotes. For example, *cond()* is unnamed and refers to any condition or disease, while *cond("hypertension")* is named as it refers to a specific condition.

Normalization using MetaMapLite can often result in high recall but low precision, as MetaMapLite has no NER component and tends to return UMLS concepts which match a given phrase syntactically but refer to abstract concepts not of interest (e.g., a search for "BMI" may return "body mass index" (C1305855), but also organic chemical "BMI 60" (C0910133)). To improve normalization precision, we employ two strategies. First, our NER component filters predicted UMLS concepts to only those of specific semantic types. For example, we limit condition concepts to only those related to semantic types of diseases or syndrome (dsyn) and so on. Next, using term-frequencies pre-computed across UMLS concept phrases, we compare term frequency-inverse document frequency (tf-idf) on MetaMapLite predictions, removing UMLS concepts whose summed matched spans have a tf-idf score lower than that of unmatched spans in a given named entity. For example, for the string "covid-19 infection", MetaMapLite predicts both "COVID-19" (C5203670) as well as several concepts related to general infections. Our tf-idf strategy removes general infection concepts because "infection" has a lower tf-idf score than the summed scores for "covid" + "-" + "19".

Laboratory values present a particular challenge, as LeafAI expects predicted lab concepts to have directly associated LOINC codes, while MetaMapLite typically normalizes lab test strings to UMLS concepts of semantic type "laboratory test or finding", but which do not have direct mappings to LOINC codes. For example, a search for "platelet count" returns the concept "Platelet Count Measurement" (C0032181), but not the needed concept of "Platelet # Bld Auto" (C0362994) with LOINC code "777-3". Thus similar to Lee and Uzuner with medications [143], we trained a BERT model for sequence classification to normalize lab tests. We trained this model to identify UMLS concepts associated with LOINC codes most frequently used in eligibility criteria [144], with each CUI as a possible class.

7.4.5 Reasoning using an integrated knowledge base

We leverage our KB (Chapter 5) in order to reason upon non-specific criteria using a combination of relatively simple rules within our API executed as SPARQL queries.

Our KB, nested logical forms, and inside-to-outside normalization methods enable "multi-hop" reasoning on eligibility criteria over several steps. For example, given the non-specific criterion "Contraindications to drugs for conditions which affect respiratory function", our system successfully reasons that (among other results),

1. **Asthma** causes changes to **respiratory function**
2. **Methylprednisolone** can be used to treat **asthma**
3. **Mycosis** (fungal infection) is a contraindication to **methylprednisolone**

These features allow LeafAI to reason upon fairly complex non-specific criteria.

7.4.6 Query generation using semantic metadata mapping

To enable data model-agnostic query generation we leverage our SMM system (Chapter 6), implemented as SQL database tables. The SMM includes metadata records of databases, tables, and columns, with corresponding foreign keys, UMLS CUI annotations, conditions, SAB encodings, and so on. The SMM is loaded from the LeafAI application database upon API startup.

7.5 Evaluation

It is reasonable to expect that an NLP-based system for finding patients based on eligibility criteria would find many patients who actually enrolled in a real clinical trial — assuming that patients enrolled in those trials met the necessary criteria as determined by study investigators. While there are caveats to this approach (for example, certain diagnosis codes may be missing for some patients, etc.), we suggest that tools such as LeafAI be evaluated by their ability to handle real-world eligibility criteria and clinical data.

We compared LeafAI's results to that of a human database programmer with over 5 years of experience as a skilled analyst using SQL databases and other tools related to our EHR. Our evaluation was performed as follows:

1. We extracted metadata on 168 clinical trials from our EHR between January 2017 and December 2021 where at least 10 patients were indicated as enrolled and not withdrawn, and the total number of raw lines of free-text within the eligibility criteria (besides the phrases "Inclusion Criteria" and "Exclusion Criteria") was less than or equal to 30.

2. By manual review, we excluded 22 trials with multiple sub-groups, as it would not be possible to know which eligibility criteria applied to which sub-group of enrolled patients.

3. To narrow the scope of our evaluation, we chose to evaluate only trials studying the following 7 disease groups: Cardiology, COVID-19, Crohn's Disease, Multiple Sclerosis (MS), Diabetes Mellitus, Hepatitis C, and Cancer. Using the "condition" field for each trial within metadata from `https://clinicaltrials.gov`, we filtered and grouped the remaining 146 trials into only those studying our diseases of interest. These diseases were selected to provide a diverse representation of potential queries that not only span different systems or medical specialties (infection, malignancy, cardiac, neurologic, endocrinologic) but also included conditions (COVID-19, Crohn's disease, diabetes) commonly studied in clinical trials.

4. We randomly chose 1 trial from each group, with the exception of Cancer, where given the large number of trials and variety of cancer types, we chose 2 trials. 427 patients were enrolled across the chosen 8 clinical trials.

5. Both LeafAI and the human programmer were provided the raw text of eligibility criteria from `https://clinicaltrials.gov`, (LeafAI using API calls). Each then created queries to find patients based on each eligibility criteria, which we executed on an OMOP database derived from our EHR containing our institution's entire research-eligible patient population.

6. To ensure results returned would be limited to only data available during the time

of each trial, we replaced references to the SQL function for generating a current timestamp (*GETDATE()*) with that of each trial's end date, and similarly replaced OMOP table references with SQL views filtering data to only that existing prior to each trial's study completion date.

7. To ensure queries would be comparable to LeafAI, the human programmer was instructed to (1) ignore criteria which cannot be computed, such as "Willing and able to participate in the study" or criteria subject to "the opinion of the investigator", as the intents and consents of patients and investigators are unavailable in our data, (2) make a best effort to reason upon non-specific criteria (e.g., symptoms for a condition), (3) not check whether patients found by a human query enrolled within a trial, and (4) skip criteria which cause an overall query to find no eligible patients. While our team did review examples of each of these cases, we did not define formal guidelines and the human programmer instead used their best judgment.

7.6 Results

Results of the query generation experiment are shown in Table 7.1. Overall, LeafAI matched 212 of 427 (49%) total enrolled patients across 8 clinical trials compared to 180 (42%) found by queries of the human programmer. The mean per-trial percent of patients matched was 43.5% for LeafAI and 27.2% for the human programmer. LeafAI had a greater number of patients determined to be eligible across all 8 trials, for a total of 27,225 eligible compared to 14,587 found by the human programmer.

Table 7.2 shows the number of criteria which were skipped by LeafAI. Of the 103 total criteria across all 8 studies, LeafAI executed queries for 61 (59.3%) and skipped 5 (4.8%) as it found no patients and 42 (40.7%) because no computable concepts were found.

Figure 7.3 shows differences in query strategies for 4 trials between LeafAI and the human programmer.

		LeafAI		Human		
Condition	**Enrolled**	**Matched**	**Eligible**	**Matched**	**Eligible**	**Time (hrs)**
CL Lymphoma	83	80 (96%)	3,252	77 (92%)	2,382	1
Hepatitis C	42	33 (78%)	9,529	32 (76%)	9,372	4
Crohn's Disease	16	0 (0%)	113	1 (6%)	9	2
Cardiac Arrest	27	12 (44%)	4,792	0 (0%)	598	5
COVID-19	41	0 (0%)	0	0 (0%)	98	2
Multiple Sclerosis	196	77 (39%)	4,891	69 (35%)	1,016	3
Type 1 Diabetes	11	0 (0%)	1,006	1 (9%)	1,104	4
Ovarian Cancer	11	10 (91%)	1,667	0 (0%)	8	5
Mean		43.5%		27.2%		
Total	427	212 (49%)	27,225	180 (42%)	14,587	26

Table 7.1: Statistics for each clinical trial evaluated by the LeafAI query engine and human programmer. The number of enrolled and matched patients were determined by cross-matching enrollments listed within our EHR. *# Crit.* indicates the number of lines of potential criteria, defined as any text besides blank spaces and the phrases "Inclusion criteria" and "Exclusion criteria".

7.7 Discussion

Our results demonstrate that LeafAI is capable of rivaling the ability of an experienced human programmer in identifying patients who are potentially eligible for clinical trials. Indeed, in numerous cases we found LeafAI and the human programmer executing similar queries, such as for Hepatitis C (NCT04852822), Chronic Lymphocytic Leukemia (NCT04852822), MS (NCT03621761), and Diabetes Mellitus (NCT03029611), where both ultimately matched a similar number of patients. 243 unique patients were matched in total by either LeafAI or the human programmer, with 149 (61.2%) identified by both.

		LeafAI			Human		
Condition	**# Crit.**	**No Pats.**	**Not Comp.**	**Exec.**	**No Pats.**	**Not Comp.**	**Exec.**
Cl Lymphoma	4	0 (0%)	0 (0%)	4 (100%)	0 (0%)	1 (25%)	3 (75%)
Hepatitis C	8	0 (0%)	4 (50%)	4 (50%)	0 (0%)	4 (50%)	4 (50%)
Crohn's Disease	9	0 (0%)	4 (44%)	5 (55%)	3 (33%)	3 (33%)	3 (33%)
Cardiac Arrest	12	0 (0%)	8 (66%)	4 (33%)	2 (16%)	5 (41%)	5 (41%)
COVID-19	13	0 (0%)	6 (46%)	7 (53%)	0 (0%)	7 (53%)	6 (46%)
Multiple Sclerosis	14	1 (7%)	3 (21%)	10 (71%)	1 (7%)	5 (35%)	8 (57%)
Type 1 Diabetes	18	2 (11%)	8 (44%)	8 (44%)	4 (22%)	6 (33%)	8 (44%)
Ovarian Cancer	25	2 (8%)	9 (36%)	14 (56%)	1 (4%)	9 (36%)	15 (60%)
Total	103	5 (5%)	42 (41%)	56 (54%)	11 (10%)	40 (39%)	52 (50%)

Table 7.2: LeafAI and the human programmer's handling of eligibility criteria for each trial. The column *No Pats.* (Patients) indicates the count of criteria which would, if executed, cause no patients to be eligible. The column *Not Computable* indicates the count of criteria which were non-computable, for various reasons. For both LeafAI and the human programmer these types of criteria were ignored. *Exec.* refers to the count to fully executed queries.

One notable pattern we found is that LeafAI consistently finds a higher number of potentially eligible patients. We hypothesize that in many cases, LeafAI's KB played a key role in this. For example, in the MS trial, LeafAI searched for 11 different SNOMED codes related to MS (including MS of the spinal cord, MS of the brain stem, acute relapsing MS, etc.), while the human programmer searched for only one, and ultimately LeafAI found nearly 5 times the number of potentially eligible patients (4,891 versus 1,016). It is possible that the human programmer had a lower rate of false positives (higher precision). However, deter-

mining this would come at the expense of manually reviewing tens of thousands of patient records to determine true eligibility and will be explored in a future analysis.

On the other hand, in the same trial, as can be seen in Figure 7.3 (A), given the exclusion criteria: "Current shift work sleep disorder, or narcolepsy diagnosed with polysomnography and multiple sleep latency", LeafAI's KB unnecessarily excluded otherwise eligible patients by removing those with diagnosis codes for drowsiness, snoring, etc., since within the UMLS those are child concepts of sleep disorder (C0851578). The exclusion of these patients likely resulted in an approximately 40% drop in recall at that stage compared to the human programmer, though ultimately both achieved similar recall (LeafAI: 39% versus Human: 35%).

Another challenging pattern we identified is the normalization of text to coded values. In the Ovarian Cancer trial (NCT03029611), both LeafAI and the human programmer matched eligible patients until line 10, which specified "Serum creatinine =< 2 or creatinine clearance > 60 ml/min...". The human programmer was unable to find a LOINC code for creatinine clearance and instead queried only for serum creatinine, finding 3 relatively rare tests which none of the enrolled patients had performed. In contrast, LeafAI normalized the serum creatinine test to LOINC code 2160-0, which 10 patients had performed. In the case of the Cardiac Arrest trial (NCT04217551), as in Figure 4 (D), in the first criterion, "Coma after resuscitation from out of hospital cardiac arrest", LeafAI attempted to create a temporal sequence query for coma diagnoses, but failed to normalize "resuscitation" and skipped the criterion as non-computable. The human programmer searched for patients with a coma diagnosis, which no enrolled patients had. In the following criterion, "Cooled to <34 deg C within 240 minutes of cardiac arrest", LeafAI searched for patients with a diagnosis of cardiac arrest, which 12 patients had. LeafAI ultimately matched 12 of 27 (44%) versus zero for the human programmer.

Beyond performance measured by recall, it is notable that the human programmer spent approximately 26 hours crafting queries for the 8 trials while LeafAI took only several minutes running on a single laptop. The time saved by using automated means such as LeafAI for cohort discovery may save health organizations significant time and resources.

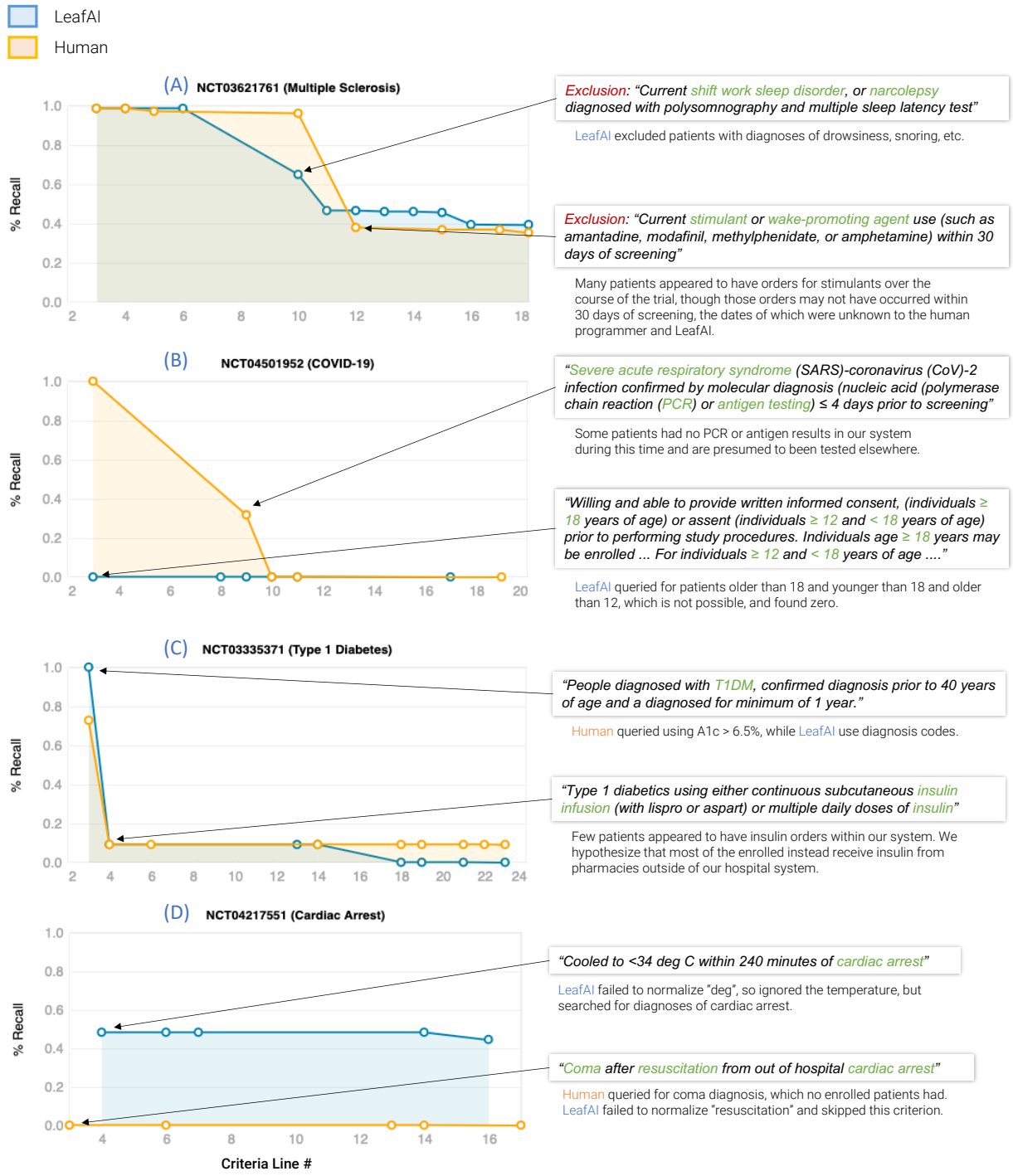


Figure 7.3: Longitudinal results for four trials. The blue line indicates recall for LeafAI and orange the human programmer. The X axis represents the line number for each eligibility criteria. Dots indicate that a query was executed for a given line. On the right, boxes represent the text of a criteria, with comments below discussing strategies and findings.

Limitations

The LeafAI query engine and our evaluation have a number of limitations. First, while the 8 clinical trials we evaluated were randomly selected, we specifically restricted the categories of diseases from which trials were chosen and limited to trials with 30 or less lines of eligibility criteria, and thus our results may not generalize to other kinds of trials. Next, we evaluated our queries using an OMOP-based extract which did not contain the full breadth of data within our EHR. Had our experiments been conducted using our enterprise data warehouse (populated by our EHR), it is possible the human programmer would have achieved better results than LeafAI due to knowledge and experience in utilizing granular source data. For example, in the Cardiac Arrest trial, the human programmer noted that data for use of cooling blankets is available in our EHR, but not in OMOP. It is not clear how LeafAI would perform were such data available. Our tests were also limited to only one institution, and it is possible that other institutions implementing different clinical trials and accessing different databases may find different results. We also did not directly compare LeafAI to other NLP systems. While we considered comparing LeafAI to Criteria2Query [6] as part of our baseline, our analysis reviewed results longitudinally (i.e., line by line of criteria), a function which Criteria2Query does not perform. As our team members are also not expert users of Criteria2Query, any direct comparison may be biased as we may not be able to use Criteria2Query appropriately for query generation.

Additionally, of the 103 criteria included in the 8 trials studied, LeafAI executed queries for only 56 (54%) and the human for 52 (50%) of them. While many criteria were unknowable (e.g., "In the opinion of investigators") or not present in our data (e.g., "Consent to the study"), others were not computable due to failures of normalization or incorrectly predicted logical form structure. While the number of skipped criteria demonstrates that improvements to LeafAI are needed, the number of criteria found non-computable by the human programmer was similar, suggesting that potential improvements along these lines may be limited. Instead, study investigators might have to be more aware of computing

shortfalls when designing criteria that will be executed by both programs and programmers; we leave a deeper analysis of this to future work. Last, the number of truly eligible patients within our institution for each trial is unknown, which impedes our ability to measure system performance. We used each trial's known enrolled patients as our gold standard, but assume they represent only a subset of those eligible. We recognize that additional analyses regarding the false positive and true negative rates are needed. These analyses were not undertaken in this study given our limited resources and the need for manual review of many thousands of patient records to complete them.

Last, we chose to evaluate by comparing query results (i.e., returned patient IDs of those meeting criteria) rather than more traditional metrics, such as BLEU or ROUGE scores, for several reasons. First, the same query result can be returned using a variety of SQL syntax approaches, potentially unfairly penalizing a given evaluation query with a lower BLEU or ROUGE score despite returning the same query result. Second, SQL query styles can vary between human programmers as well, again potentially penalizing machine-generated approaches based on the arbitrary stylistic preferences of a given programmer used as a gold standard. More complex database schema and longer queries exacerbate both of these challenges. Last, even if these syntax and stylistic challenges were not present, we argue that the results of a query are more important regardless. One can imagine, for example, cases where a gold standard and evaluation query differ only by a single character, such as

"SELECT patient_id FROM patients WHERE Age $\geq$ *18"*

"SELECT patient_id FROM patients WHERE Age $<$ *18"*

With the difference of only the inequality operator ($\geq$ versus $<$) the evaluation query has a relatively high BLEU score of 70.1%, despite returning zero correct patients! Evaluating instead using the returned results would return zero patients matched and thus 0% recall. For these reasons, we found comparison of query results rather than syntax to be a more meaningful and useful metric.

7.8 Conclusions

This chapter introduced LeafAI, a NLP-based system leveraging deep learning and an integrated KB which can automatically generate queries for cohort discovery on virtually any clinical data model. Using an OMOP database representing the entire patient population of our institution, we demonstrated that LeafAI rivals the performance of an experienced human programmer in identifying eligible research candidates.

Chapter 8
WEB APPLICATION

8.1 Overview

This chapter presents the goals and development of the LeafAI web application, an interactive user-facing web interface which integrates with the LeafAI API described in Chapter 7. As development of the web application is in progress, we leave evaluation of this to future work. Section 8.2 describes related work. Section 8.3 describes methods. Section 8.4 discusses future work, including user testing, and Section 8.5 concludes this chapter.

8.2 Related Work

Experimental systems enabling the querying of databases using natural language interfaces have been in development since the 1960s. Using relatively simple rule-based parsing systems, Woods [145] created a system for asking natural language questions of a moon rock database, while Epstein and Walker [146] similarly designed a natural language interface for a melanoma database. Decades later, Katz *et al* created START [147], a system capable of basic question-answering using data extracted and parsed from the internet. In the biomedical informatics domain, Cao *et al* developed AskHERMES [148], question-answering software capable of answering medical questions related to drugs, contraindications, and so on, using support vector machines (SVMs) and an internal knowledge base derived from the UMLS.

Cohort discovery systems using natural language are in many ways a subset of systems for question-answering which answer only one (unstated) question, "How many patients meet these criteria?". As discussed, in the biomedical informatics and clinical trials domain, the most well-known and cited system for matching patients to clinical trials using free-text

eligibility criteria is Criteria2Query [6, 8]. Criteria2Query offers a web-based simple and friendly user interface for inputting free-text eligibility criteria and enables users to correct erroneously normalized named entities.

8.3 Methods

The web application forms one component of the LeafAI application, which is a 3-tier architecture similar to Leaf [2]: a back-end with clinical and application databases, and server hosting an API (Chapter 7), and a user-facing web application. The web application features a chat-like design.

A chat-like interface for cohort discovery is both novel and a logical design choice given the natural language interface of LeafAI used for query generation. An example of LeafAI's user interface is shown in Figure 8.1. As can be seen, we assume that the order of user-provided criteria is intentional and important, and leverage that assumption to both structure queries incrementally to report results line by line, with each reported result (e.g., "421 are aged between 18 and 65") effectively a subset of the preceding result.

The web application responds to user input in a visual form that is chat-*like*, rather than the result of a general programmatic conversation agent. While the user interface and query generation methods lay a potential foundation for general-purpose question-answering more akin to a conversation agent, we limit our scope to cohort discovery. The user interface was designed with the following goals:

1. **Accessible history**. Users will be able to immediately scroll to view previous findings.

2. **Rapid feedback and explainability**. The application performs NER and normalization as users type. This allows users to preemptively detect concepts and queries which may return unexpected results. The application also returns incremental query results in real-time, providing immediate feedback so users may avoid waiting until all query results are complete.

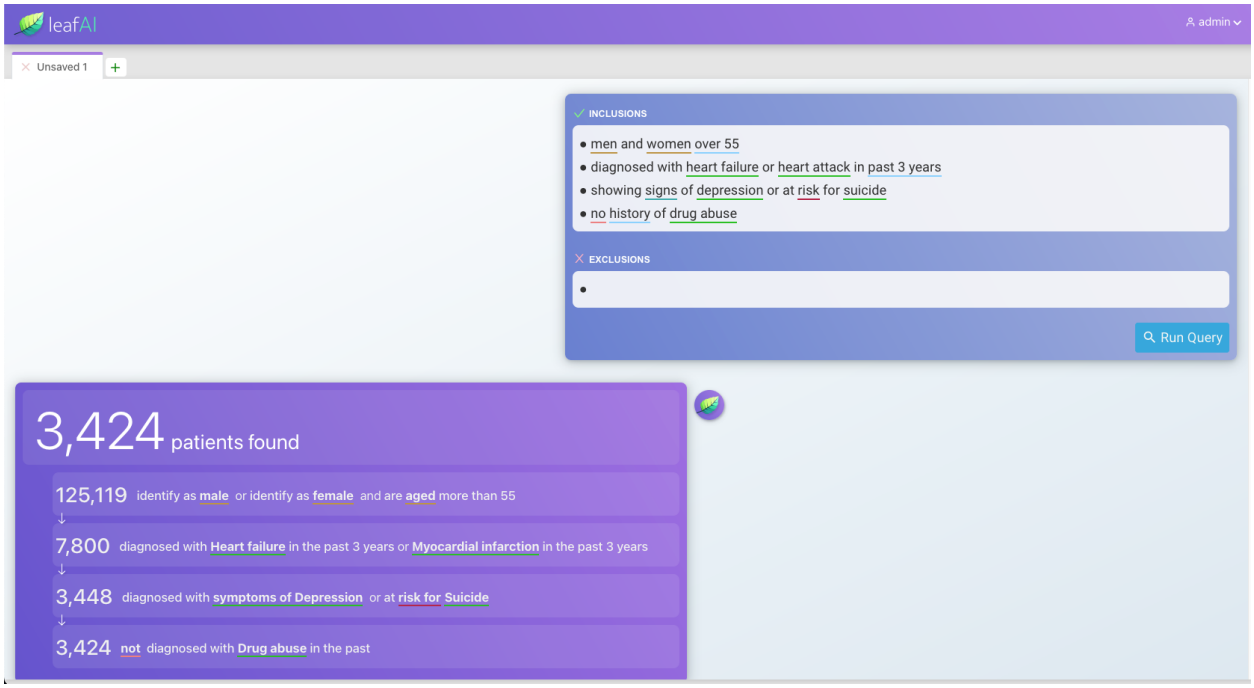


Figure 8.1: The web application user interface. User-entered criteria are shown in the above right, while system responses are shown in the lower left. User criteria are displayed and executed in order, with each count of patients representing a subset of the preceding count.

3. **Direct editing of responses enabling iteration**. As the application returns results of patients found, the eligibility criteria used in a query will be directly edit-able, saving users' time and facilitating quick iteration to find intended patients.

We next describe each goal in detail.

8.3.1 Accessible History

Cohort discovery is a form of data exploration. As Derthick and Roth write, "...the data exploration process is not characterized by monotonic progress towards a goal, but rather involves much backtracking and opportunistic goal revision" [149]. Put another way, user goals and perceptions may change over the course of exploration. With ubiquitous vertical

scrolling - where more recent actions and utterances are inserted downward while history is preserved upward - chat-like user interfaces facilitate user understanding of past utterances and actions. Persisted, easily viewable history of user utterances and actions enable what Gergle *et al* call "conversational grounding" [150], that is, accessible data to guide users to previously acquired findings and information. This history of previous actions can improve the pace of discovery and alleviate the need for users to (often imperfectly) attempt to recall their earlier findings and paths taken [151, 150].

8.3.2 Rapid Feedback and explainability

Users' sense of system latency and responsiveness can significantly affect their satisfaction in using a tool [152, 153, 154]. Faster *preemptive* system responses (i.e., informing users' of a possible consequence before they complete an action) can also both save users' time and reduce loads placed upon systems by preventing unnecessary actions [155, 156]. The web application employs two general strategies for providing rapid feedback to users, one before queries are executed, and the second while results are being reported during query execution.

Consider a hypothetical case where a user begins to type a criteria but misspells "Diabetes Mellitus". Given no notification of the error, the user may take seconds or even minutes of additional typing while adding new criteria without realizing the initial spelling mistake. After the user awaits LeafAI's response, she finds that the query found no patients and is frustrated and confused at the counter-intuitive result, only to finally notice the misspelling, having wasted several minutes. The application avoids this scenario by preemptively performing NER and normalization while users are typing. Examples of this are shown in Figure 8.2. Named entities found within user criteria are underlined and interactive, enabling users to better estimate whether their queries will succeed or not.

Figure 8.2: Normalized named entities identified in user input before query execution. On the left, the system correctly identified "BMI". On the right, "Diabetes Mellitus" is misspelled, and the user is notified.

Second, as LeafAI generates incremental queries and returns results line by line asynchronously, the user interface shows results as they are reported using a streaming interface. As a result, users do not need to wait until all queries are complete to obtain visual feedback as to query results (as in tools such as Leaf and i2b2). An example of this is shown in Figure 8.3.

Taken together, these features for rapid feedback and transparency also enable *explainability* of system actions. Rather than simply returning a final count of patients meeting criteria, The user interface provides information both before and during query execution of how the system interpreted user intent and how it has executed a query (e.g. what concepts it found or did not, misinterpreted, etc.). System responses are generated using templated English expressions mapped to logical form types using normalized UMLS concept names. By avoiding being a "black box", the user interface is designed to gain user trust and understanding of how the system operates.

Figure 8.3: A time-lapse representation asynchronously reporting of incremental query results, from top to bottom. This is performed using two streaming interfaces, one from the clinical database to the API, and a second from the API to the web application.

8.3.3 Direct editing of responses enabling iteration

Data exploration is iterative: users explore, try, and learn over the course of multiple attempts. As discussed, faster, meaningful results also directly affect user satisfaction. To this end, the web application enables users to immediately and directly edit the returned system responses. This workflow is depicted in Figure 8.4.

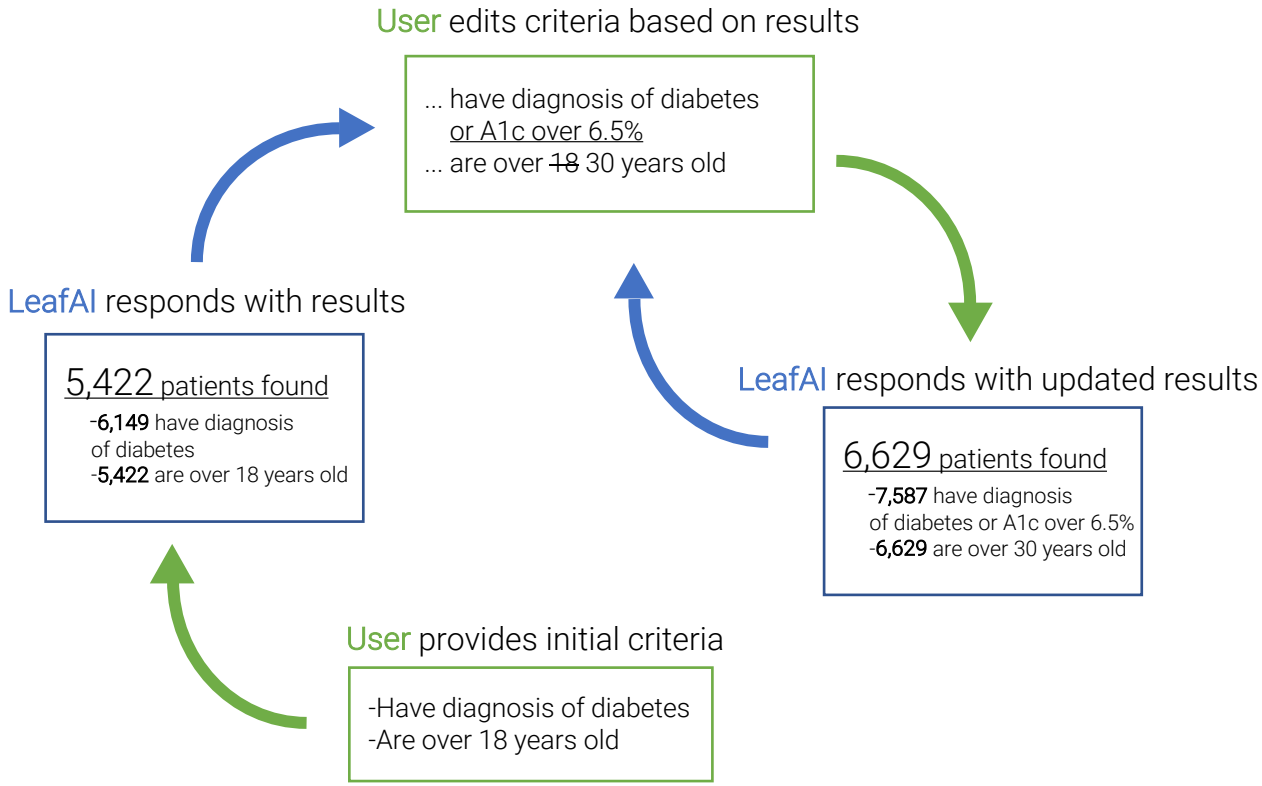


Figure 8.4: An iterative workflow using an example case for adults with Diabetes Mellitus. Users are able to directly edit results and re-execute their queries while preserving query history, saving user time and preserving previous user actions and findings.

For example, after seeing that LeafAI found less patients than expected, a user may realize that she should slightly alter her original query to expand her search. Rather than needing to copy and paste her earlier criteria, instead she is able to simply click and modify

the earlier results.

Additionally, as discussed, the LeafAI query engine is capable of reasoning upon non-specific criteria. In certain cases, however, the system's findings may be incomplete, incorrect, or undesired for various reasons. Rather than forcing users to accept imperfect reasoning, however, the user interface allows users to directly enable or disable reasoned concepts. An example of this is shown in Figure 8.5. In addition to reasoned concepts, the interface enables users to enable and disable named entities from system responses and immediately re-execute queries with a single click, shown in Figure 8.6. This allows for fast iteration and saves user time, obviating the need to alter the original inputs.

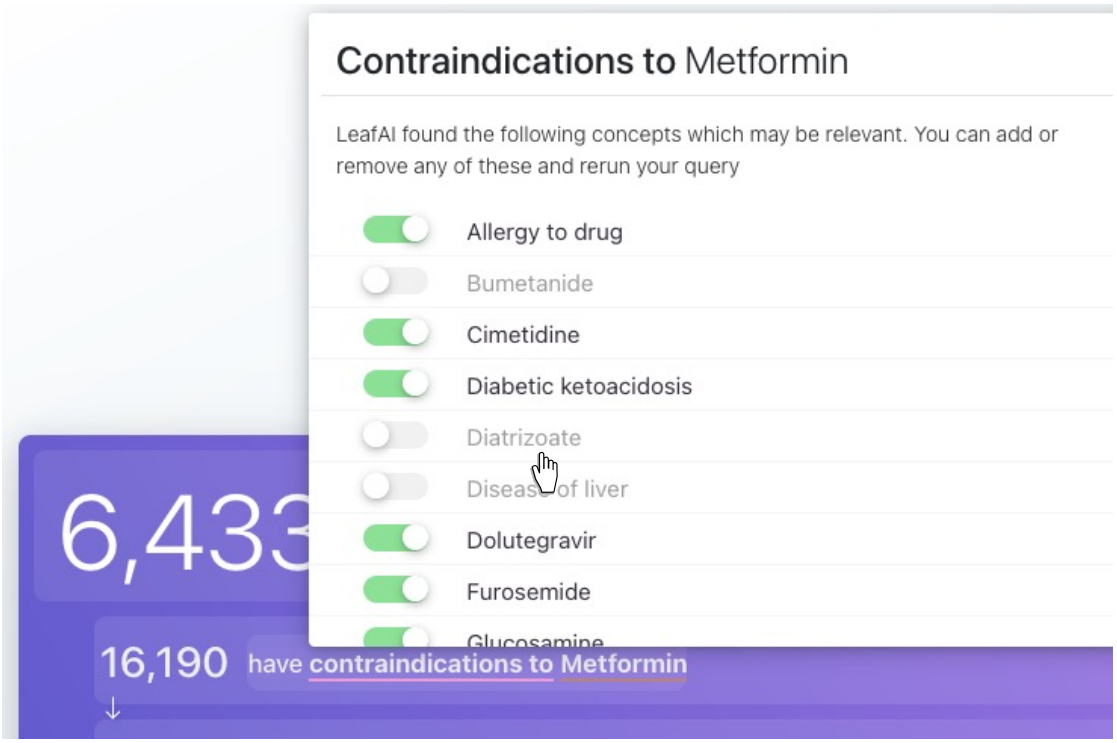


Figure 8.5: An example of a user editing concepts discovered using LeafAI's reasoning system. Users are able to enable or disable reasoned concepts.

The system's reasoning can thus be described as an optional means of helpfully saving users' time, which can be accepted, edited, or discarded as needed.

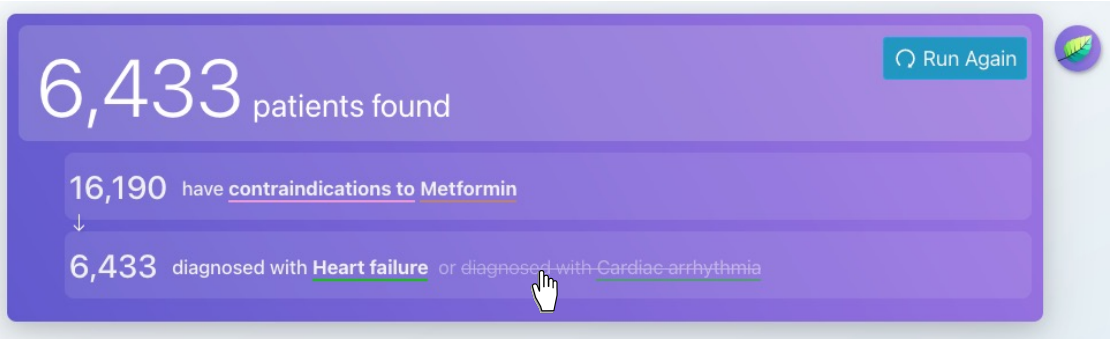


Figure 8.6: An example of a user disabling a named entity from a system response. Named entities can be enabled and disabled by clicking directly on them. Disabled entities are shown with gray colored text and struck through by a line. Upon re-execution, logic for disabled entities is removed from the generated query.

8.3.4 Model Inference Speed

LeafAI's modules for NER, logical form transformation, lab normalization and so on use large Transformer-based neural models [157] with millions of parameters. Using these models as-is for inference can be slow, particularly when performed using CPUs rather than more costly but often far faster GPUs (Graphics Processing Units). Delays in inference time in turn can result in poor system latency, affecting user satisfaction. We further assume that institutions or individuals deploying LeafAI may not necessarily have access to GPUs. To improve inference speed on CPUs, models are quantized, a process for converting 32-bit floating point values within model weights and biases to 8-bit integers. Quantization has been shown to dramatically reduce model storage size and memory usage and improve inference speed while typically showing limited decreases in performance [158].

8.4 Future Work

In future work we will employ a usability and user satisfaction and acceptability study. We hypothesize that the innovative design of the web application user interface will represent

a meaningful advance in user interface design for cohort discovery. Future versions of the application may allow for general question-answering, data visualization, and collaborative data analysis and query authorship between users of the system and LeafAI.

8.5 Conclusion

This chapter described the ongoing development of the LeafAI web application, which incorporates a number of innovative user-facing features such as streaming interface updates during query execution, real-time named entity recognition and normalization, as well as mixed-modal interaction allowing users to manipulate system responses and re-execute generated queries, enabling the system to act as a form of augmented intelligence for a user. In future work we will evaluate the web application in a usability and user satisfaction and acceptability study.

Chapter 9
CONCLUSIONS

This dissertation describes the development of a novel database query generation system to identify patients eligible for clinical trials using a natural language interface. We conclude with a summary of the primary contributions of this work, a discussion of limitations, and finally remarks on potential directions for future research.

9.1 Contributions

Recruitment of eligible patients for clinical trials remains a costly and time-consuming challenge, and contributes in many cases to slowing the process of bringing new drugs and treatment options to patients who may benefit from them. The process of identifying patients who may qualify for clinical trial is also challenging, as manual chart review is highly labor intensive, interactive tools such as Leaf are not always able to represent appropriate queries to find patients in EHRs, and the technical expertise needed to run database queries to find patients directly from an EHR database is often a limited resource. Applications leveraging natural language descriptions of eligibility criteria to automatically sift through data and find patients therefore hold great potential to simplify and expedite this process.

It is our hope that the research and software applications described in this dissertation are impactful for real-world clinical trials. We summarize our primary contributions here:

1. *Annotated Corpora*: we introduced two new human-annotated corpora, the Leaf Clinical Trials (LCT) corpus and Leaf Logical Forms (LLF) corpus. The LCT corpus was developed using a highly granular named entity- and relation-based schema to enable robust query generation for real-world clinical trials and eligibility criteria. The corpus consists of over 1,000 documents of eligibility criteria from clinical trials. Our best

performing NER prediction baseline model trained on the corpus achieved an F_1 score of 81.3%, while our best performing relation extraction model achieved an F_1 score of 85.2%. The LLF corpus builds upon the LCT corpus by representing 2,000 eligibility criteria lines with corresponding logical intermediate representations, or logical forms. We developed the LLF corpus annotation schema with the aim of creating a data model-agnostic, flexible, robust, and machine-readable representation of the underlying logic of a given criteria. Our best performing model for predicting a logical form from a free-text input achieved a BLEU score of 93.5%.

2. *Model-agnostic query generation using metadata*: academic medical centers and research institutions use a variety of clinical database systems and schema. While research-oriented common data models such as OMOP are increasingly used to enable data and tool sharing and cross-institutional analysis, the research database landscape remains heterogeneous. Moreover, many institutions and projects alter common data models in order to accommodate specific project and analysis needs. Because of this, we developed a novel data model-agnostic query generation method enabled by "tagging" clinical database schema metadata using a UMLS concepts in what we call a Semantic Metadata Mapping, or SMM. Our SMM system enables great flexibility in representing a variety of data models, including relatively complex common data models such as OMOP.

3. *Knowledge Base*: a natural language interface for identifying cohorts using eligibility criteria requires an external knowledge representation system because humans often use non-specific phrasing and references for brevity and convenience. We developed a graph-based Knowledge Base, or KB, by integrating a wide variety of publicly available data sets, ontologies, and vocabulary mapping systems, with the Unified Medical Language System, or UMLS, at its core.

4. *Reasoning upon non-specific criteria*: Indirect criterion which frequently appear in

eligibility criteria, such as contraindications to a drug, symptoms of a condition, or risks of an outcome, must be reasoned upon in order to determine patients meeting a given criterion. Leveraging our KB and logical forms, we demonstrate a novel inside-to-outside sequential reasoning method which allows our system to generate queries meeting non-specific criteria.

5. *Evaluation against an experienced human programmer and actual trial enrollments*: of the natural language interfaces for eligibility criteria developed to date, to our knowledge only one used actual participant enrollments as part of a benchmark, and none compared system performance to that of an experienced human clinical database programmer. In contrast, we evaluated our system's generated queries to that of a human programmer, setting a new and higher benchmark for this task. We demonstrate that our system is able to rival—and by some measures surpass—a human programmer in matching patients found by generated queries to those who actually enrolled, with our system identifying 212 of 427 (49%) total enrolled patients across 8 clinical trials compared to 180 (42%) found by queries of the human programmer.

6. *Natural language interfaces to databases*: the ability to ask questions of a database using non-technical language has been a goal of researchers for decades. This work contributes to this body of research, demonstrating that a hybrid system using neural networks, a KB, metadata and relatively simple rules can be capable of querying a complex clinical database in a relatively sophisticated manner.

7. *An interactive web application interface*: data exploration, including cohort discovery, is an iterative process. We designed a user-friendly chat-like web application and user interface to enable users to interact with our system and find patients meeting criteria using a natural language interface. Our user interface incorporates a number of novel features to provide immediate user feedback and explainability, including real-time query results for each line of criteria, NER and normalization as users type, and direct

manipulation and re-execution of query results.

9.2 Limitations

The research described in this dissertation has a number of limitations, many of which have been described in preceding chapters. The corpora we introduce, the LCT and LLF corpora, were analyzed in great detail to ensure quality, but were nevertheless annotated by non-clinicians, and as is the case of any human-annotated corpus, may contain errors. Models trained on these corpora may also return erroneous output and thus inaccurate queries downstream. Our KB, while containing a wide variety of sources and thus capable of enabling reasoning across many demonstrated use cases, nevertheless lacks functionality to embed probabilities, significance, or caveats to its output, and thus may return erroneous results. We believe these shortcomings are acceptable given that users of our web application may edit any reasoned concepts.

In our query generation task, it is not clear how well our system may generalize to other kinds of clinical trials outside of the 8 we tested on. Perhaps most importantly, our evaluation underscores the challenges in measuring system performance when there exists no true gold standard of clinical trial eligibilities, particularly across an entire institutional data warehouse. We use known participant enrollments in actual clinical trials but recognize the shortcomings of this approach.

For the web application, as discussed, we have not yet evaluated whether our system meets or exceeds usability and user acceptance standards, nor how well our system compares to previous user-facing systems.

9.3 Future Work

There are a number of promising avenues for future research which may build upon our findings and systems. We first look forward to testing and improving our web application and query generation methods with robust user testing. Like other self-service tools, it is in the hands of users that the impact of such systems will be made.

One future research direction may be the exploration of enabling partially-matched patients for cohort discovery, as opposed to the existing relatively simple per-patient Boolean approach. In certain cases, a researcher may be interested in finding patients who meet a majority of criteria, but not necessarily all. This may be particularly useful in cases of studies of rare diseases, and one can imagine a useful system which suggests alterations to criteria in order to find additional eligible patients. In a related fashion, LeafAI's use of logical forms may be adapted to automatically detect logical inconsistencies and notify users, such as for patients simultaneously older and younger than a given age.

Another possible direction is the adaption of our system for general-purpose question-answering or dynamic data visualization. After identifying a cohort of interest, it is natural for users to ask, "Of these patients, what were their outcomes post-surgery?", or, "Did any show signs of X in the following year?", or, "Show a comparison of their demographics alongside a control cohort". One can imagine such capabilities being of potential value in expanding the analytic capabilities of tools such as LeafAI.

An additional future project may be the creation of a de-identified data set of structured clinical data (and possibly unstructured notes as well) of patients alongside human expert-annotated indications of whether they were eligible or not for a given set of clinical trials. While the manual review required to determine whether patients were eligible or not would not be trivial, the practical value to the research community in improving tools such as our own and objectively comparing approaches would be significant.

BIBLIOGRAPHY

[1] R. L. Richesson, W. E. Hammond, M. Nahm, D. Wixted, G. E. Simon, J. G. Robinson, A. E. Bauck, D. Cifelli, M. M. Smerek, J. Dickerson, *et al.*, "Electronic health records based phenotyping in next-generation clinical trials: a perspective from the NIH Health Care Systems Collaboratory," *Journal of the American Medical Informatics Association*, vol. 20, no. e2, pp. e226–e231, 2013.

[2] N. J. Dobbins, C. H. Spital, R. A. Black, J. M. Morrison, B. de Veer, E. Zampino, R. D. Harrington, B. D. Britt, K. A. Stephens, A. B. Wilcox, P. Tarczy-Hornoch, and S. D. Mooney, "Leaf: an open-source, model-agnostic, data-driven web application for cohort discovery and translational biomedical research," *Journal of the American Medical Informatics Association*, vol. 27, pp. 109–118, 10 2019.

[3] S. N. Murphy, G. Weber, M. Mendis, V. Gainer, H. C. Chueh, S. Churchill, and I. Kohane, "Serving the enterprise and beyond with informatics for integrating biology and the bedside (i2b2)," *Journal of the American Medical Informatics Association*, vol. 17, no. 2, pp. 124–130, 2010.

[4] E. K. Johnson, S. Broder-Fingert, P. Tanpowpong, J. Bickel, J. R. Lightdale, and C. P. Nelson, "Use of the i2b2 research query tool to conduct a matched case–control clinical research study: advantages, disadvantages and methodological considerations," *BMC medical research methodology*, vol. 14, no. 1, pp. 1–6, 2014.

[5] V. G. Deshmukh, S. M. Meystre, and J. A. Mitchell, "Evaluating the informatics for integrating biology and the bedside system for clinical research," *BMC medical research methodology*, vol. 9, no. 1, pp. 1–12, 2009.

[6] C. Yuan, P. B. Ryan, C. Ta, Y. Guo, Z. Li, J. Hardin, R. Makadia, P. Jin, N. Shang, T. Kang, and C. Weng, "Criteria2Query: A natural language interface to clinical databases for cohort definition," *Journal of the American Medical Informatics Association*, vol. 26, no. 4, pp. 294–305, 2019.

[7] S. Soni and K. Roberts, "Patient cohort retrieval using transformer language models," in *AMIA annual symposium proceedings*, vol. 2020, p. 1150, American Medical Informatics Association, 2020.

[8] Y. Fang, B. Idnay, Y. Sun, H. Liu, Z. Chen, K. Marder, H. Xu, R. Schnall, and C. Weng, "Combining human and machine intelligence for clinical trial eligibility querying," *Journal of the American Medical Informatics Association*, 2022.

[9] X. Zhang, C. Xiao, L. M. Glass, and J. Sun, "Deepenroll: patient-trial matching with deep embedding and entailment prediction," in *Proceedings of The Web Conference 2020*, pp. 1029–1037, 2020.

[10] L. Chen, Y. Gu, X. Ji, C. Lou, Z. Sun, H. Li, Y. Gao, and Y. Huang, "Clinical trial cohort selection based on multi-level rule-based natural language processing system," *Journal of the American Medical Informatics Association*, vol. 26, no. 11, pp. 1218–1226, 2019.

[11] D. F. Patrão, M. Oleynik, F. Massicano, and A. Morassi Sasso, "Recruit-an ontology based information retrieval system for clinical trials recruitment," in *MEDINFO 2015: eHealth-enabled Health*, pp. 534–538, IOS Press, 2015.

[12] H. Dhayne, R. Kilany, R. Haque, and Y. Taher, "Emr2vec: Bridging the gap between patient data and clinical trial," *Computers & Industrial Engineering*, vol. 156, p. 107236, 2021.

[13] R. Liu, S. Rizzo, S. Whipple, N. Pal, A. L. Pineda, M. Lu, B. Arnieri, Y. Lu, W. Capra, R. Copping, *et al.*, "Evaluating eligibility criteria of oncology trials using real-world data and ai," *Nature*, vol. 592, no. 7855, pp. 629–633, 2021.

[14] Y. Xiong, X. Shi, S. Chen, D. Jiang, B. Tang, X. Wang, Q. Chen, and J. Yan, "Cohort selection for clinical trials using hierarchical neural network," *Journal of the American Medical Informatics Association*, vol. 26, no. 11, pp. 1203–1208, 2019.

[15] A. Y. Wang, W. J. Lancaster, M. C. Wyatt, L. V. Rasmussen, D. G. Fort, and J. J. Cimino, "Classifying clinical trial eligibility criteria to facilitate phased cohort identification using clinical data repositories," in *AMIA Annual Symposium Proceedings*, vol. 2017, p. 1754, American Medical Informatics Association, 2017.

[16] J. Ross, S. Tu, S. Carini, and I. Sim, "Analysis of eligibility criteria complexity in clinical trials," *Summit on translational bioinformatics*, vol. 2010, p. 46, 2010.

[17] A. Sertkaya, H.-H. Wong, A. Jessup, and T. Beleche, "Key cost drivers of pharmaceutical clinical trials in the united states," *Clinical Trials*, vol. 13, no. 2, pp. 117–126, 2016.

[18] N. J. Dobbins, T. Mullen, Ö. Uzuner, and M. Yetisgen, "The leaf clinical trials corpus: a new resource for query generation from clinical trial eligibility criteria," *Scientific Data*, vol. 9, no. 1, pp. 1–15, 2022.

[19] O. Bodenreider, "The unified medical language system (UMLS): integrating biomedical terminology," *Nucleic acids research*, vol. 32, no. suppl_1, pp. D267–D270, 2004.

[20] L. M. Schriml, C. Arze, S. Nadendla, Y.-W. W. Chang, M. Mazaitis, V. Felix, G. Feng, and W. A. Kibbe, "Disease ontology: a backbone for disease semantic integration," *Nucleic acids research*, vol. 40, no. D1, pp. D940–D946, 2012.

[21] E. W. Sayers, T. Barrett, D. A. Benson, E. Bolton, S. H. Bryant, K. Canese, V. Chetvernin, D. M. Church, M. DiCuccio, S. Federhen, *et al.*, "Database resources of the national center for biotechnology information," *Nucleic acids research*, vol. 39, no. suppl_1, pp. D38–D51, 2010.

[22] A. Sargsyan, A. T. Kodamullil, S. Baksi, J. Darms, S. Madan, S. Gebel, O. Keminer, G. M. Jose, H. Balabin, L. N. DeLong, *et al.*, "The covid-19 ontology," *Bioinformatics*, vol. 36, no. 24, pp. 5703–5705, 2020.

[23] S. Ayvaz, J. Horn, O. Hassanzadeh, Q. Zhu, J. Stan, N. P. Tatonetti, S. Vilar, M. Brochhausen, M. Samwald, M. Rastegar-Mojarad, *et al.*, "Toward a complete dataset of drug–drug interaction information from publicly available sources," *Journal of biomedical informatics*, vol. 55, pp. 206–217, 2015.

[24] X. A. Zhang, A. Yates, N. Vasilevsky, J. Gourdine, T. J. Callahan, L. C. Carmody, D. Danis, M. P. Joachimiak, V. Ravanmehr, E. R. Pfaff, *et al.*, "Semantic integration of clinical laboratory tests from electronic health records for deep phenotyping and biomarker discovery," *NPJ digital medicine*, vol. 2, no. 1, pp. 1–9, 2019.

[25] X. Wang, A. Chused, N. Elhadad, C. Friedman, and M. Markatou, "Automated knowledge acquisition from clinical narrative reports," in *AMIA Annual Symposium Proceedings*, vol. 2008, p. 783, American Medical Informatics Association, 2008.

[26] "Icd-9-cm to and from icd-10-cm and icd-10-pcs crosswalk or general equivalence mappings." `https://www.nber.org/research/data/icd-9-cm-and-icd-10-cm-and-icd-10-pcs-crosswalk-or-general-equivalence-mappings`. Accessed: 2022-08-16.

[27] "Icd-9-cm diagnostic codes to snomed ct map." `https://www.nlm.nih.gov/research/umls/mapping_projects/icd9cm_to_snomedct.html`. Accessed: 2022-08-16.

[28] "Icd-9-cm procedure codes to snomed ct map." https://www.nlm.nih.gov/research/umls/mapping_projects/icd9cmv3_to_snomedct.html. Accessed: 2022-08-16.

[29] "Snomed ct to icd-10-cm map." SNOMEDCTtoICD-10-CMMap/umls/mapping_projects/icd9cmv3_to_snomedct.html. Accessed: 2022-08-16.

[30] L. M. Friedman, C. D. Furberg, D. L. DeMets, D. M. Reboussin, and C. B. Granger, *Fundamentals of clinical trials*. Springer, 2015.

[31] P. M. Rothwell, "External validity of randomised controlled trials:"to whom do the results of this trial apply?"," *The Lancet*, vol. 365, no. 9453, pp. 82–93, 2005.

[32] G. Frank, "Current challenges in clinical trial patient recruitment and enrollment," *SoCRA Source*, vol. 2, no. February, pp. 30–38, 2004.

[33] J. D. Grill and J. Karlawish, "Addressing the challenges to successful recruitment and retention in alzheimer's disease clinical trials," *Alzheimer's research & therapy*, vol. 2, no. 6, pp. 1–11, 2010.

[34] R. B. Gul and P. A. Ali, "Clinical trials: the challenge of recruitment and retention of participants," *Journal of clinical nursing*, vol. 19, no. 1-2, pp. 227–233, 2010.

[35] C. Heller, J. E. Balls-Berry, J. D. Nery, P. J. Erwin, D. Littleton, M. Kim, and W. P. Kuo, "Strategies addressing barriers to clinical trial enrollment of underrepresented populations: a systematic review," *Contemporary clinical trials*, vol. 39, no. 2, pp. 169–182, 2014.

[36] M. Adams, L. Caffrey, and C. McKevitt, "Barriers and opportunities for enhancing patient recruitment and retention in clinical research: findings from an interview study in an nhs academic health science centre," *Health research policy and systems*, vol. 13, no. 1, pp. 1–9, 2015.

[37] R. D. Nipp, K. Hong, and E. D. Paskett, "Overcoming barriers to clinical trial enrollment," *American Society of Clinical Oncology Educational Book*, vol. 39, pp. 105–114, 2019.

[38] B. A. Guadagnolo, D. G. Petereit, P. Helbig, D. Koop, P. Kussman, E. Fox Dunn, and A. Patnaik, "Involving american indians and medically underserved rural populations in cancer clinical trials," *Clinical trials*, vol. 6, no. 6, pp. 610–617, 2009.

[39] L. Penberthy, R. Brown, F. Puma, and B. Dahman, "Automated matching software for clinical trials eligibility: measuring efficiency and flexibility," *Contemporary clinical trials*, vol. 31, no. 3, pp. 207–217, 2010.

[40] D. R. Holmes, J. Major, D. E. Lyonga, R. S. Alleyne, and S. M. Clayton, "Increasing minority patient participation in cancer clinical trials using oncology nurse navigation," *The American journal of surgery*, vol. 203, no. 4, pp. 415–422, 2012.

[41] J. Sullivan, "Subject recruitment and retention: barriers to success," 2004.

[42] S. R. Thadani, C. Weng, J. T. Bigger, J. F. Ennever, and D. Wajngurt, "Electronic screening improves efficiency in clinical trial recruitment," *Journal of the American Medical Informatics Association*, vol. 16, no. 6, pp. 869–873, 2009.

[43] P. Easterbrook and D. Matthews, "Fate of research studies.," *Journal of the Royal Society of Medicine*, vol. 85, no. 2, p. 71, 1992.

[44] A. M. McDonald, R. C. Knight, M. K. Campbell, V. A. Entwistle, A. M. Grant, J. A. Cook, D. R. Elbourne, D. Francis, J. Garcia, I. Roberts, *et al.*, "What influences recruitment to randomised controlled trials? a review of trials funded by two uk funding agencies," *Trials*, vol. 7, no. 1, pp. 1–8, 2006.

[45] L. Marks and E. Power, "Using technology to address recruitment issues in the clinical trial process," *Trends in biotechnology*, vol. 20, no. 3, pp. 105–109, 2002.

[46] Y. Ni, S. Kennebeck, J. W. Dexheimer, C. M. McAneney, H. Tang, T. Lingren, Q. Li, H. Zhai, and I. Solti, "Automated clinical trial eligibility prescreening: increasing the efficiency of patient identification for clinical trials in the emergency department," *Journal of the American Medical Informatics Association*, vol. 22, no. 1, pp. 166–178, 2015.

[47] P. A. Harris, R. Taylor, R. Thielke, J. Payne, N. Gonzalez, and J. G. Conde, "Research electronic data capture (redcap)—a metadata-driven methodology and workflow process for providing translational research informatics support," *Journal of biomedical informatics*, vol. 42, no. 2, pp. 377–381, 2009.

[48] M. Sok, M. Zavrl, B. Greif, and M. Srpčič, "Objective assessment of who/ecog performance status," *Supportive Care in Cancer*, vol. 27, no. 10, pp. 3793–3798, 2019.

[49] S. M. Lundberg, B. Nair, M. S. Vavilala, M. Horibe, M. J. Eisses, T. Adams, D. E. Liston, D. K.-W. Low, S.-F. Newman, J. Kim, *et al.*, "Explainable machine-learning predictions for the prevention of hypoxaemia during surgery," *Nature biomedical engineering*, vol. 2, no. 10, pp. 749–760, 2018.

[50] E. Jermutus, D. Kneale, J. Thomas, and S. Michie, "Influences on user trust in healthcare artificial intelligence: A systematic review," *Wellcome Open Research*, vol. 7, p. 65, 2022.

[51] H. S. Dar, M. I. Lali, M. U. Din, K. M. Malik, and S. A. C. Bukhari, "Frameworks for querying databases using natural language: a literature review," *arXiv preprint arXiv:1909.01822*, 2019.

[52] G. Hripcsak, J. D. Duke, N. H. Shah, C. G. Reich, V. Huser, M. J. Schuemie, M. A. Suchard, R. W. Park, I. C. K. Wong, P. R. Rijnbeek, *et al.*, "Observational Health Data Sciences and Informatics (OHDSI): opportunities for observational researchers," *Studies in health technology and informatics*, vol. 216, p. 574, 2015.

[53] S. Bae, D. Kim, J. Kim, and E. Choi, "Question answering for complex electronic health records database using unified encoder-decoder architecture," in *Machine Learning for Health*, pp. 13–25, PMLR, 2021.

[54] J. Park, Y. Cho, H. Lee, J. Choo, and E. Choi, "Knowledge graph-based question answering with electronic health records," in *Machine Learning for Healthcare Conference*, pp. 36–53, PMLR, 2021.

[55] P. Wang, T. Shi, and C. K. Reddy, "Text-to-sql generation for question answering on electronic medical records," in *Proceedings of The Web Conference 2020*, pp. 350–361, 2020.

[56] Y. Pan, C. Wang, B. Hu, Y. Xiang, X. Wang, Q. Chen, J. Chen, J. Du, *et al.*, "A bert-based generation model to transform medical texts to sql queries for electronic medical records: Model development and validation," *JMIR Medical Informatics*, vol. 9, no. 12, p. e32698, 2021.

[57] A. E. Johnson, T. J. Pollard, L. Shen, L.-w. H. Lehman, M. Feng, M. Ghassemi, B. Moody, P. Szolovits, L. Anthony Celi, and R. G. Mark, "Mimic-iii, a freely accessible critical care database," *Scientific data*, vol. 3, no. 1, pp. 1–9, 2016.

[58] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[59] "Apache lucene." `https://lucene.apache.org/`. Accessed: 2022-08-16.

[60] "Owl 2 web ontology language primer." `https://www.w3.org/TR/owl2-primer/`. Accessed: 2022-08-16.

[61] C. Patel, J. Cimino, J. Dolby, A. Fokoue, A. Kalyanpur, A. Kershenbaum, L. Ma, E. Schonberg, and K. Srinivas, "Matching patient records to clinical trials using ontologies," in *The Semantic Web*, pp. 816–829, Springer, 2007.

[62] S. Tu, M. Peleg, S. Carini, D. Rubin, and I. Sim, "Ergo: A templatebased expression language for encoding eligibility criteria," tech. rep., Technical report, 2009.

[63] Z. Huang, A. t. Teije, and F. v. Harmelen, "Semanticct: a semantically-enabled system for clinical trials," in *Process Support and Knowledge Representation in Health Care*, pp. 11–25, Springer, 2013.

[64] F. Baader, S. Borgwardt, and W. Forkel, "Patient selection for clinical trials using temporalized ontology-mediated query answering," in *Companion Proceedings of the The Web Conference 2018*, pp. 1069–1074, 2018.

[65] C. Weng, X. Wu, Z. Luo, M. R. Boland, D. Theodoratos, and S. B. Johnson, "EliXR: an approach to eligibility criteria extraction and representation," *Journal of the American Medical Informatics Association*, vol. 18, pp. i116–i124, dec 2011.

[66] M. R. Boland, S. W. Tu, S. Carini, I. Sim, and C. Weng, "EliXR-TIME: A Temporal Knowledge Representation for Clinical Research Eligibility Criteria.," *AMIA Joint Summits on Translational Science proceedings. AMIA Joint Summits on Translational Science*, vol. 2012, pp. 71–80, 2012.

[67] T. Kang, S. Zhang, Y. Tang, G. W. Hruby, A. Rusanov, N. Elhadad, and C. Weng, "EliIE: An open-source information extraction system for clinical trial eligibility criteria," *Journal of the American Medical Informatics Association*, vol. 24, no. 6, pp. 1062–1071, 2017.

[68] C. Sutton, A. McCallum, *et al.*, "An introduction to conditional random fields," *Foundations and Trends® in Machine Learning*, vol. 4, no. 4, pp. 267–373, 2012.

[69] F. Kury, A. Butler, C. Yuan, L.-h. Fu, Y. Sun, H. Liu, I. Sim, S. Carini, and C. Weng, "Chia, a large annotated corpus of clinical trial eligibility criteria," *Scientific data*, vol. 7, no. 1, pp. 1–11, 2020.

[70] Y. Sun, A. Butler, L. A. Stewart, H. Liu, C. Yuan, C. T. Southard, J. H. Kim, and C. Weng, "Building an omop common data model-compliant annotated corpus for covid-19 clinical trials," *Journal of biomedical informatics*, vol. 118, p. 103790, 2021.

[71] X. Yu, T. Chen, Z. Yu, H. Li, Y. Yang, X. Jiang, and A. Jiang, "Dataset and Enhanced Model for Eligibility Criteria-to-SQL Semantic Parsing," no. May, pp. 5829–5837, 2020.

[72] N. Guarino and P. Giaretta, "Ontologies and knowledge bases," *Towards very large knowledge bases*, pp. 1–2, 1995.

[73] J. Lehmann, R. Isele, M. Jakob, A. Jentzsch, D. Kontokostas, P. N. Mendes, S. Hellmann, M. Morsey, P. Van Kleef, S. Auer, *et al.*, "Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia," *Semantic web*, vol. 6, no. 2, pp. 167–195, 2015.

[74] N. F. Noy, N. H. Shah, P. L. Whetzel, B. Dai, M. Dorf, N. Griffith, C. Jonquet, D. L. Rubin, M.-A. Storey, C. G. Chute, *et al.*, "Bioportal: ontologies and integrated data resources at the click of a mouse," *Nucleic acids research*, vol. 37, no. suppl_2, pp. W170–W173, 2009.

[75] D. Martinez, A. Otegi, A. Soroa, and E. Agirre, "Improving search over electronic health records using umls-based query expansion through random walks," *Journal of biomedical informatics*, vol. 51, pp. 100–106, 2014.

[76] J. Sankhavara, R. Dave, B. Dave, and P. Majumder, "Query specific graph-based query reformulation using umls for clinical information access," *Journal of Biomedical Informatics*, vol. 108, p. 103493, 2020.

[77] S. Schulz and U. Hahn, "Medical knowledge reengineering—converting major portions of the umls into a terminological knowledge base," *International Journal of Medical Informatics*, vol. 64, no. 2-3, pp. 207–221, 2001.

[78] H. Kazi, B. S. Chowdhry, and Z. Memon, "Medchatbot: an umls based chatbot for medical students," 2012.

[79] X. V. Lin, R. Socher, and C. Xiong, "Multi-hop knowledge graph reasoning with reward shaping," *arXiv preprint arXiv:1808.10568*, 2018.

[80] B. Hao, H. Zhu, and I. C. Paschalidis, "Enhancing clinical bert embedding using a biomedical knowledge base," in *28th International Conference on Computational Linguistics (COLING 2020)*, 2020.

[81] K.-H. Huang, M. Yang, and N. Peng, "Biomedical event extraction with hierarchical knowledge graphs," *arXiv preprint arXiv:2009.09335*, 2020.

[82] F. Petroni, T. Rocktäschel, P. Lewis, A. Bakhtin, Y. Wu, A. H. Miller, and S. Riedel, "Language models as knowledge bases?," *arXiv preprint arXiv:1909.01066*, 2019.

[83] A. X. Chang and C. D. Manning, "Sutime: A library for recognizing and normalizing time expressions.," in *Lrec*, vol. 3735, p. 3740, 2012.

[84] P. Stenetorp, S. Pyysalo, G. Topić, T. Ohta, S. Ananiadou, and J. Tsujii, "Brat: a web-based tool for nlp-assisted text annotation," in *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 102–107, 2012.

[85] F. Dernoncourt, J. Y. Lee, and P. Szolovits, "NeuroNER: an easy-to-use program for named-entity recognition based on neural networks," *arXiv preprint arXiv:1705.05487*, 2017.

[86] I. Beltagy, K. Lo, and A. Cohan, "Scibert: A pretrained language model for scientific text," *arXiv preprint arXiv:1903.10676*, 2019.

[87] Y. Gu, R. Tinn, H. Cheng, M. Lucas, N. Usuyama, X. Liu, T. Naumann, J. Gao, and H. Poon, "Domain-specific language model pretraining for biomedical natural language processing," *ACM Transactions on Computing for Healthcare (HEALTH)*, vol. 3, no. 1, pp. 1–23, 2021.

[88] S. Wu and Y. He, "Enriching pre-trained language model with entity information for relation classification," in *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pp. 2361–2364, 2019.

[89] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 1532–1543, 2014.

[90] A. Kamath and R. Das, "A survey on semantic parsing," in *Automated Knowledge Base Construction (AKBC)*.

[91] J. Herzig, P. Shaw, M.-W. Chang, K. Guu, P. Pasupat, and Y. Zhang, "Unlocking compositional generalization in pre-trained models using intermediate representations," *arXiv preprint arXiv:2104.07478*, 2021.

[92] K. Roberts and D. Demner-Fushman, "Annotating logical forms for ehr questions," in *LREC... International Conference on Language Resources & Evaluation:[proceedings]. International Conference on Language Resources and Evaluation*, vol. 2016, p. 3772, NIH Public Access, 2016.

[93] M. A. Musen, S. W. Tu, A. K. Das, and Y. Shahar, "Eon: A component-based approach to automation of protocol-directed therapy," *Journal of the American Medical Informatics Association*, vol. 3, no. 6, pp. 367–388, 1996.

[94] S. W. Tu, J. R. Campbell, J. Glasgow, M. A. Nyman, R. McClure, J. McClay, C. Parker, K. M. Hrabak, D. Berg, T. Weida, *et al.*, "The sage guideline model: achievements and overview," *Journal of the American Medical Informatics Association*, vol. 14, no. 5, pp. 589–598, 2007.

[95] I. Sim, B. Olasov, and S. Carini, "An ontology of randomized controlled trials for evidence-based practice: content specification and evaluation using the competency decomposition method," *Journal of Biomedical Informatics*, vol. 37, no. 2, pp. 108–119, 2004.

[96] R. K. Hulse, S. J. Clark, J. C. Jackson, H. R. Warner, and R. M. Gardner, "Computerized medication monitoring system," *American Journal of Hospital Pharmacy*, vol. 33, no. 10, pp. 1061–1070, 1976.

[97] M. A. Musen, "The protégé project: a look back and a look forward," *AI matters*, vol. 1, no. 4, pp. 4–12, 2015.

[98] M. Sordo, A. A. Boxwala, O. Ogunyemi, and R. A. Greenes, "Description and status update on gello: a proposed standardized object-oriented expression language for clinical decision support," in *MEDINFO 2004*, pp. 164–168, IOS Press, 2004.

[99] S. Miksch, Y. Shahar, and P. Johnson, "Asbru: a task-specific, intention-based, and time-oriented language for representing skeletal plans," in *Proceedings of the 7th Workshop on Knowledge Engineering: Methods & Languages (KEML-97)*, pp. 9–19, Milton Keynes, UK, The Open University, Milton Keynes, UK, 1997.

[100] M. J. O'Connor, S. W. Tu, and M. A. Musen, "The chronus ii temporal database mediator.," in *Proceedings of the AMIA Symposium*, p. 567, American Medical Informatics Association, 2002.

[101] C. Weng, S. W. Tu, I. Sim, and R. Richesson, "Formal representation of eligibility criteria: a literature review," *Journal of biomedical informatics*, vol. 43, no. 3, pp. 451–467, 2010.

[102] S. Soni, S. Datta, and K. Roberts, "quehry: a question answering system to query electronic health records," *Journal of the American Medical Informatics Association*, p. ocad050, 2023.

[103] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, P. J. Liu, *et al.*, "Exploring the limits of transfer learning with a unified text-to-text transformer.," *J. Mach. Learn. Res.*, vol. 21, no. 140, pp. 1–67, 2020.

[104] A. Einolghozati, P. Pasupat, S. Gupta, R. Shah, M. Mohit, M. Lewis, and L. Zettlemoyer, "Improving semantic parsing for task oriented dialog," *arXiv preprint arXiv:1902.06000*, 2019.

[105] S. Rongali, L. Soldaini, E. Monti, and W. Hamza, "Don't parse, generate! a sequence to sequence architecture for task-oriented semantic parsing," in *Proceedings of The Web Conference 2020*, pp. 2962–2968, 2020.

[106] C.-Y. Lin, "Rouge: A package for automatic evaluation of summaries," in *Text summarization branches out*, pp. 74–81, 2004.

[107] C. Callison-Burch, M. Osborne, and P. Koehn, "Re-evaluating the role of bleu in machine translation research," in *11th conference of the european chapter of the association for computational linguistics*, pp. 249–256, 2006.

[108] F. Manola, E. Miller, B. McBride, *et al.*, "Rdf primer," *W3C recommendation*, vol. 10, no. 1-107, p. 6, 2004.

[109] C. Friedman, P. O. Alderson, J. H. Austin, J. J. Cimino, and S. B. Johnson, "A general natural-language text processor for clinical radiology," *Journal of the American Medical Informatics Association*, vol. 1, no. 2, pp. 161–174, 1994.

[110] F. Priyatna, O. Corcho, and J. Sequeda, "Formalisation and experiences of r2rml-based sparql to sql query translation using morph," in *Proceedings of the 23rd international conference on World wide web*, pp. 479–490, 2014.

[111] O. Inc., "Graphdb." `https://www.ontotext.com/products/graphdb/`. Accessed: 2023-05-04.

[112] D. Beckett and Berners-Lee, "Rdf 1.1 turtle." `https://www.w3.org/TR/turtle//`. Accessed: 2023-05-04.

[113] S. Harris and A. Seaborne, "Sparql 1.1 query language." `https://www.w3.org/TR/sparql11-query/`. Accessed: 2023-05-04.

[114] R. Belenkaya, M. J. Gurley, A. Golozar, D. Dymshyts, R. T. Miller, A. E. Williams, S. Ratwani, A. Siapos, V. Korsik, J. Warner, *et al.*, "Extending the omop common data model and standardized vocabularies to support observational cancer research," *JCO Clinical Cancer Informatics*, vol. 5, 2021.

[115] Y. Peng, A. Nassirian, N. Ahmadi, M. Sedlmayr, and F. Bathelt, "Towards the representation of genomic data in hl7 fhir and omop cdm.," in *GMDS*, pp. 86–94, 2021.

[116] M. Zoch, C. Gierschner, Y. Peng, M. Gruhl, L. A. Leutner, M. Sedlmayr, F. Bathelt, *et al.*, "Adaption of the omop cdm for rare diseases.," in *MIE*, pp. 138–142, 2021.

[117] J. L. Warner, D. Dymshyts, C. G. Reich, M. J. Gurley, H. Hochheiser, Z. H. Moldwin, R. Belenkaya, A. E. Williams, and P. C. Yang, "Hemonc: A new standard vocabulary for chemotherapy regimen representation in the omop common data model," *Journal of biomedical informatics*, vol. 96, p. 103239, 2019.

[118] X. Zhou, S. Murugesan, H. Bhullar, Q. Liu, B. Cai, C. Wentworth, and A. Bate, "An evaluation of the thin database in the omop common data model for active drug safety surveillance," *Drug safety*, vol. 36, no. 2, pp. 119–134, 2013.

[119] S. J. Shin, S. C. You, Y. R. Park, J. Roh, J.-H. Kim, S. Haam, C. G. Reich, C. Blacketer, D.-S. Son, S. Oh, *et al.*, "Genomic common data model for seamless interoperation of biomedical data in clinical practice: retrospective study," *Journal of medical Internet research*, vol. 21, no. 3, p. e13249, 2019.

[120] E. Kwon, C.-W. Jeong, D. Kang, Y. Kim, Y. Lee, K.-H. Yoon, *et al.*, "Development of common data module extension for radiology data (r-cdm): A pilot study to predict outcome of liver cirrhosis with using portal phase abdominal computed tomography data," European Congress of Radiology-ECR 2019, 2019.

[121] T. Berners-Lee, J. Hendler, and O. Lassila, "The semantic web," *Scientific american*, vol. 284, no. 5, pp. 34–43, 2001.

[122] C. Bizer and A. Seaborne, "Treating non-rdf databases as virtual rdf graphs,"

[123] J. F. Sequeda, S. H. Tirmizi, O. Corcho, and D. P. Miranker, "Direct mapping sql databases to the semantic web: A survey," *Univeristy of Texas, Department of Computer Sciences Technical Report TR-09-04*, p. 83, 2009.

[124] J. F. Sequeda, S. H. Tirmizi, O. Corcho, and D. P. Miranker, "Survey of directly mapping sql databases to the semantic web," *The Knowledge Engineering Review*, vol. 26, no. 4, pp. 445–486, 2011.

[125] C. A. Knoblock, P. Szekely, J. L. Ambite, A. Goel, S. Gupta, K. Lerman, M. Muslea, M. Taheriyan, and P. Mallick, "Semi-automatically mapping structured sources into the semantic web," in *The Semantic Web: Research and Applications: 9th Extended Semantic Web Conference, ESWC 2012, Heraklion, Crete, Greece, May 27-31, 2012. Proceedings 9*, pp. 375–390, Springer, 2012.

[126] S. Gupta, P. Szekely, C. A. Knoblock, A. Goel, M. Taheriyan, and M. Muslea, "Karma: A system for mapping structured sources into the semantic web," in *The Semantic Web: ESWC 2012 Satellite Events: ESWC 2012 Satellite Events, Heraklion, Crete, Greece, May 27-31, 2012. Revised Selected Papers*, pp. 430–434, Springer, 2015.

[127] M. R. Cowie, J. I. Blomster, L. H. Curtis, S. Duclaux, I. Ford, F. Fritz, S. Goldman, S. Janmohamed, J. Kreuzer, M. Leenay, *et al.*, "Electronic health records to facilitate clinical research," *Clinical Research in Cardiology*, vol. 106, no. 1, pp. 1–9, 2017.

[128] C. H. Lee and H.-J. Yoon, "Medical big data: promise and challenges," *Kidney research and clinical practice*, vol. 36, no. 1, p. 3, 2017.

[129] J.-C. Chen, "Dijkstra's shortest path algorithm," *Journal of formalized mathematics*, vol. 15, no. 9, pp. 237–247, 2003.

[130] T. O. H. D. Sciences and I. program, "Usagi." `https://github.com/OHDSI/Usagi`. Accessed: 2023-05-05.

[131] T. O. H. D. Sciences and I. program, "Atlas." `https://github.com/OHDSI/Atlas`. Accessed: 2023-05-05.

[132] D. Bender and K. Sartipi, "Hl7 fhir: An agile and restful approach to healthcare information exchange," in *Proceedings of the 26th IEEE international symposium on computer-based medical systems*, pp. 326–331, IEEE, 2013.

[133] T. L. Foundation, "grpc: A high performance, open source universal rpc framework." `https://grpc.io//`. Accessed: 2022-08-16.

[134] D. Inc., "Docker." `https://www.docker.com`. Accessed: 2022-08-16.

[135] A. Johnstone and E. Scott, "Generalised recursive descent parsing and follow-determinism," in *International Conference on Compiler Construction*, pp. 16–30, Springer, 1998.

[136] A. R. Aronson, "Effective mapping of biomedical text to the umls metathesaurus: the metamap program.," in *Proceedings of the AMIA Symposium*, p. 17, American Medical Informatics Association, 2001.

[137] D. Demner-Fushman, W. J. Rogers, and A. R. Aronson, "Metamap lite: an evaluation of a new java implementation of metamap," *Journal of the American Medical Informatics Association*, vol. 24, no. 4, pp. 841–844, 2017.

[138] A. B. Abacha, A. Seco De Herrera, S. Gayen, D. Demner-Fushman, and S. Antani, "Nlm at imageclef 2017 caption task," in *Working Notes of CLEF 2017-Conference and Labs of the Evaluation Forum (CLEF 2017), Dublin, Ireland, September 11-14, 2017.*, vol. 1866, CEUR Workshop Proceedings, 2017.

[139] C. Liu, C. N. Ta, J. R. Rogers, Z. Li, J. Lee, A. M. Butler, N. Shang, F. S. P. Kury, L. Wang, F. Shen, *et al.*, "Ensembles of natural language processing systems for portable phenotyping solutions," *Journal of biomedical informatics*, vol. 100, p. 103318, 2019.

[140] H. Wang, Y. Li, S. A. Khan, and Y. Luo, "Prediction of breast cancer distant recurrence using natural language processing and knowledge-guided convolutional neural network," *Artificial intelligence in medicine*, vol. 110, p. 101977, 2020.

[141] C. Liu, F. S. Peres Kury, Z. Li, C. Ta, K. Wang, and C. Weng, "Doc2hpo: a web application for efficient and accurate hpo concept curation," *Nucleic acids research*, vol. 47, no. W1, pp. W566–W570, 2019.

[142] B. G. Patra, V. Maroufy, B. Soltanalizadeh, N. Deng, W. J. Zheng, K. Roberts, and H. Wu, "A content-based literature recommendation system for datasets to improve data reusability–a case study on gene expression omnibus (geo) datasets," *Journal of Biomedical Informatics*, vol. 104, p. 103399, 2020.

[143] K. Lee and Ö. Uzuner, "Normalizing adverse events using recurrent neural networks with attention," *AMIA Summits on Translational Science Proceedings*, vol. 2020, p. 345, 2020.

[144] A. Rafee, S. Riepenhausen, P. Neuhaus, A. Meidt, M. Dugas, and J. Varghese, "Elapro, a loinc-mapped core dataset for top laboratory procedures of eligibility screening for clinical trials," *BMC Medical Research Methodology*, vol. 22, no. 1, pp. 1–14, 2022.

[145] W. A. Woods, "Progress in natural language understanding: an application to lunar geology," in *Proceedings of the June 4-8, 1973, national computer conference and exposition*, pp. 441–450, 1973.

[146] M. N. Epstein and D. E. Walker, "Natural language access to a melanoma data base," in *The Second Annual Symposium on Computer Application in Medical Care, 1978. Proceedings.*, pp. 320–325, IEEE, 1978.

[147] B. Katz, D. Yuret, J. Lin, S. Felshin, R. Schulman, A. Ilik, A. Ibrahim, and P. Osafo-Kwaako, "Integrating web resources and lexicons into a natural language query system," in *Proceedings IEEE International Conference on Multimedia Computing and Systems*, vol. 2, pp. 255–261, IEEE, 1999.

[148] Y. Cao, F. Liu, P. Simpson, L. Antieau, A. Bennett, J. J. Cimino, J. Ely, and H. Yu, "Askhermes: An online question answering system for complex clinical questions," *Journal of biomedical informatics*, vol. 44, no. 2, pp. 277–288, 2011.

[149] M. Derthick and S. F. Roth, "Enhancing data exploration with a branching history of user operations," *Knowledge-Based Systems*, vol. 14, no. 1-2, pp. 65–74, 2001.

[150] D. Gergle, D. R. Millen, R. E. Kraut, and S. R. Fussell, "Persistence matters: Making the most of chat in tightly-coupled work," in *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 431–438, 2004.

[151] W. C. Hill and J. D. Hollan, "History-enriched digital objects: Prototypes and policy issues," *The Information Society*, vol. 10, no. 2, pp. 139–145, 1994.

[152] S. Li and C.-H. Chen, "The effects of visual feedback designs on long wait time of mobile application user interface," *Interacting with Computers*, vol. 31, no. 1, pp. 1–12, 2019.

[153] I. Arapakis, X. Bai, and B. B. Cambazoglu, "Impact of response latency on user behavior in web search," in *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, pp. 103–112, 2014.

[154] B. Shneiderman, "Response time and display rate in human performance with computers," *ACM Computing Surveys (CSUR)*, vol. 16, no. 3, pp. 265–285, 1984.

[155] R. Lempel and S. Moran, "Predictive caching and prefetching of query results in search engines," in *Proceedings of the 12th international conference on World Wide Web*, pp. 19–28, 2003.

[156] F. Diaz, Q. Guo, and R. W. White, "Search result prefetching using cursor movement," in *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, pp. 609–618, 2016.

[157] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[158] I. Hubara, M. Courbariaux, D. Soudry, R. El-Yaniv, and Y. Bengio, "Quantized neural networks: Training neural networks with low precision weights and activations," *The Journal of Machine Learning Research*, vol. 18, no. 1, pp. 6869–6898, 2017.