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

Nicholas J. Dobbins
Department of Biomedical Informatics and Medical Education
School of Medicine, University of Washington
Seattle, Washington. United States

Committee Members
Meliha Yetisgen, PhD (Chair)
H. Nina Kim, MD, MSc
Trevor Cohen, MBChB, PhD
Fei Xia, PhD (Graduate School Representative)

General Exam Date: November 14, 2022

Contents

1 Specific Aims

Clinical trials serve a critical role in the generation of medical evidence and furthering of 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 [1], and manual chart review of hundreds or thousands of patients to determine a candidate pool can be prohibitively labor- and time-intensive. While cohort discovery tools such as Leaf [2] or i2b2 [3] can serve to assist in finding participants meeting eligibility criteria, such tools nonetheless often have significant learning curves, and certain complex queries may simply be impossible due to structural limitations on the types of possible queries presented.

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

The goal of this project is the development of an application called LeafAI, which if successful will enable the discovery of patients meeting criteria for clinical trials and general purpose biomedical research using natural language and generating queries for virtually any clinical database structure.

1.1 Aim 1. Creation of Gold Standard Clinical Trials Data Set

In Aim 1 we seek to create a gold standard human-annotated data set of clinical trial eligibility criteria. Our annotation schema expands on previous work [4, 5] while capturing significantly greater semantic granularity in order to facilitate accurate query generation and enable handling of complex and non-specific criteria. Predictive models trained on the resulting corpus enable subsequent work in Aim 2.

1.2 Aim 2. Query Generation Method Development and Evaluation

A successful application for cohort discovery using NLP for end-users hinges on development of robust, explainable methods for generating database queries. In Aim 2 we focus on creation of a novel database model-agnostic approach for cohort discovery query generation using NLP, including an integrated knowledge base (KB), Sequence to Sequence- (Seq2Seq) based logical transformations of eligibility criteria, normalization, and semantic metadata mapping (SMM) of database structure using concepts within the Unified Medical Language System (UMLS). To evaluate our system, we analyze queries generated for 8 randomly chosen past clinical trials at our institution and compared enrolled patients to those found our systems compared to a human programmer.

1.3 Aim 3. Web Application Development and Evaluation

Aim 3 will focus on the development of a web application enabling automatic query generation using user-entered free-text eligibility criteria. The LeafAI web application will utilize a chat-like interface to enable iterative, explainable, interactive query generation. Users will then be able to edit and re-execute queries based on their findings. We will evaluate the effectiveness of the web application by comparing both (1) usability evaluations and (2) performance of generated queries in finding enrolled patients between LeafAI and our previous non-NLP based cohort discovery tool, Leaf.

2 Background and Significance

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 [6]. 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 [6].

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* [7]. 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 [8, 9, 10, 11]. The reasons for this are varied, including overly restrictive inclusion criteria [9], a lack of awareness on the part of patients, particularly in underserved and historically disadvantaged communities [10], fear or apprehension of medical research due to past abuses [8], and uncertainty of risk on the part of providers leading to withheld offers to participate [11]. 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 [9, 10, 11, 12, 13, 14]. Beyond questions of generalizability, recruitment challenges also cause delays to clinical trials, with an estimate 86% of trials delayed between 1 and 6 months, and some for even longer [15, 16].

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. [17, 13, 18, 19].

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 [13, 16]. For example, Thadani *et al* found electronic screening methods to significantly reduce the burden of manual chart review in one study by approximately 81% [16]. 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. 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 [20].

Thus 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 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. Thus however imperfect (e.g., a lack of diagnosis data for an existing condition may cause certain patients to be inappropriately deemed ineligible), clinical trials are thus 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 are challenging for NLP systems. For example, an NLP system may correctly determine that "breast cancer" refers to a condition and "surgery" refers to a procedure, but may still fail if "Newly" is not determined to refer to breast cancer or not determined to mean something occurring for the first time. In a subsequent step, an NLP system may normalize (i.e., determine an 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

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

such. In criterion 4, an NLP system may fail to reason that other unstated conditions, such as hysteria or 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 [21] 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 by 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 [22, 23], but also challenging as many systems tend to prioritize performance over interpretability.

2.5 Broader Impact

Despite these challenges, clear opportunities exist for NLP-based systems to improve recruitment rates and efficiencies in clinical trials. This project aims to create a user-friendly, explainable, and generalizable application for cohort discovery from natural language by querying virtually any database schema. If successful, the outcome of this work will be a system capable of state-of-the-art query generation enabling significantly faster discovery of eligible patients for clinical trials and biomedical research in general.

Clinical trials are considered the gold standard of evaluation of new treatments. Yet clinical trials are often 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 [24]. Much of those costs are due to difficulties in finding and keeping patients 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.

Nor are potential impacts of this project limited to clinical trials. Biomedical research in general very often relies on finding patients meeting certain criteria, whether for retrospective 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.

3 Task Innovation

Much of the work to date on cohort discovery and RCT eligibility criteria matching has been impressive and impactful in leveraging advances in related research domains. Yet critical gaps remain in terms of practical usability, generalizability, and measurement of performance for these methods in the context of real-world clinical trials and data.

Among existing clinical trials corpora (to be discussed in Aim 1), Chia is the largest and most notable, but also can be difficult to directly translate into SQL queries as its entities and relations often require additional downstream parsing and normalization. Chia also lacks entities and concepts important to clinical trials, such as contraindications or means of distinguishing between specified and non-specifc criteria.

Among tools and methods for SQL query generation (to be discussed in Aim 2), most efforts to date are capable of generating database queries on only a single database schema, such as OMOP or MIMIC. While the OMOP database schema is widely used in research, this lack of flexibility and adaptability toward other data models limits potential utility, in particular given the widely documented necessity to change and extend the standard OMOP schema to accommodate real-world project needs [25, 26, 27, 28, 29, 30, 31]. Moreover, most methods for generating SQL queries, particularly those using Encoder-Decoder

Transformer-based architectures, tend to generate relatively simple SQL statements, with few JOINs or nested sub-queries and typically no support for UNION operators and so on.

Methods utilizing clinical notes for document ranking show great potential, particularly given the significant amount of untapped information present in free-text as compared to structured data [32]. However the use of these systems at enterprise scale or as ad hoc query tools for researchers has been limited, and the number of notes used for experiments tend to be few (hundreds or low thousands) as compared to the tens of millions of clinical notes stored within many EHRs.

Efforts using patient and eligibility criteria embeddings, while novel and showing future potential, also have notable limitations. For example, one study assumed that patients who did not enroll in a given trial were not eligible [33] (which may not necessarily be true), and none of the research we are aware of provided sufficient detail on how structured data were transformed into embeddings or what specific data elements were used, preventing direct reproducibility. Research in methods using Description Logics and related representations and ontologies, meanwhile, has been largely experimental and untested using large real-world clinical databases. Many works also require domain experts to first manually translate eligibility criteria to logical representations, making them unrealistic as cohort discovery tools. Few of the methods described provide support for complex logic, and none support reasoning on non-specific criteria (e.g., "diseases that affect respiratory function"), two phenomena common to eligibility criteria [34, 35]. Perhaps most importantly, to the best of our knowledge, only one previous work has been tested in terms of matching patients enrolled in actual clinical trials [33] (with caveats discussed earlier), and none have been directly compared to the capabilities of a human database programmer.

Among NLP-based cohort discovery tools (to be discussed in Aims 2 and 3), Criteria2Query, the most well known, offers a friendly but simplistic user interface with no support for saving queries, reasoning, or summary of patients found at each step.

This project will be highly innovative by

1. Releasing a uniquely granular and practical NLP corpus of annotated eligibility criteria.
2. Establishing a robust new state-of-the-art system for eligibility criteria matching on clinical databases of any data model with multi-hop reasoning and logical form transformation.
3. Creating a novel web-based application using a chat-based user interface which enables system interpretations line-by-line and allows iterative dynamic queries using natural language.

4 Research Plan

4.1 Aim 1: Creation of Gold Standard Clinical Trials Data Set

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 Aim 1 we aim to create a gold standard corpus of human-annotated clinical trial eligibility criteria. Predictive models trained on this corpus enable and are used in subsequent Aims of this project. This aim is complete and was published in August 2022 [2].

4.1.1 Related Work

A number of corpora related to clinical trials have been published [4, 5, 36, 37]. Weng *et al* developed EliXR [36], a rule-based information extraction (IE) pipeline and corpus of 1,000 eligibility criteria documents for NER and relation extraction. The corpus was not made publicly available. Kang *et al* created an annotation schema based on the Observational Medical Outcomes Partnership (OMOP) Common Data Model [38] and an annotated corpus of 230 eligibility criteria documents, though the corpus focused narrowly on Alzheimer's Disease-related trials only [5]. More recently, Kury *et al* created Chia [4], a publicly available corpus of 1,000 Phase IV trials similarly focused on the OMOP Common Data model but across a variety of disease domains and with a greater number of types of entities and relations. Yu *et al* [37] 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 is not clear this approach is yet viable for real-world clinical trials recruitment.

4.1.2 Eligibility Criteria and Database Queries

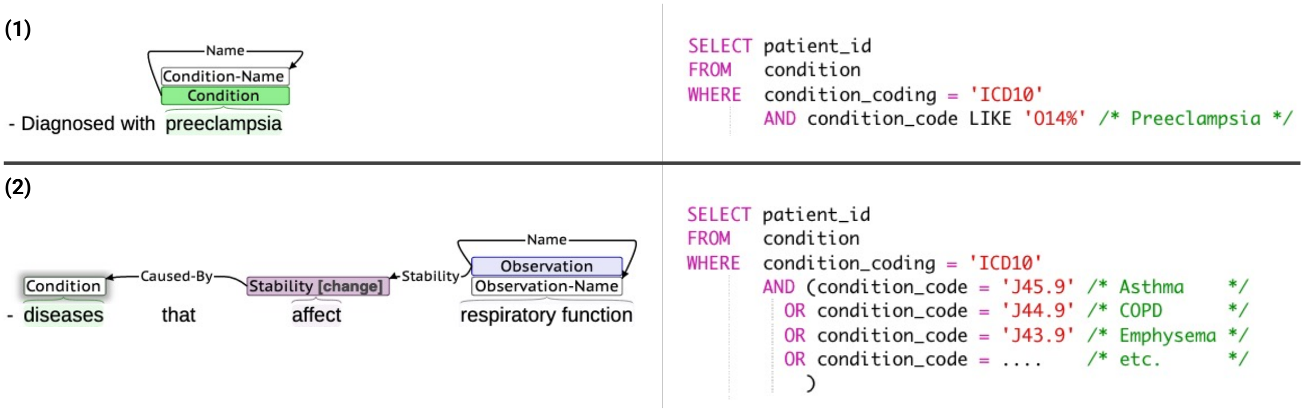
The NLP tasks involved in transforming eligibility criteria into database queries include **named entity recognition** (NER) to tag meaningful spans of text as named entities, **relation extraction** to classify relations between named entities, **normalization** to map named entities to common coded representations (e.g., ICD-10), **negation detection** to detected 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. Figure 1 illustrates why corpora structure and integrity are important for the task of query generation, using examples of eligibility criteria annotated using the LCT 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 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 LCT annotation schema in order to enable reasoning and ease query generation for real-world clinical trials use. As the second example in Figure 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.

4.1.3 Annotation schema

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 [36, 40, 5, 4] 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.

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.



```
SELECT patient_id
FROM   condition
WHERE  condition_coding = 'ICD10'
       AND condition_code LIKE 'O14%' /* Preeclampsia */
```

```
SELECT patient_id
FROM   condition
WHERE  condition_coding = 'ICD10'
       AND (condition_code = 'J45.9' /* Asthma    */
         OR condition_code = 'J44.9' /* COPD      */
         OR condition_code = 'J43.9' /* Emphysema */
         OR condition_code = ....    /* etc.      */
           )
```

Figure 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 [39]. The ICD-10 codes shown are examples and not intended to be exhaustive.

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 programmatic 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. 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.*
- **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.

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 (↓) to denote entity annotation and left and right arrow symbols (← and →) 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:

$$\begin{array}{ccc} \text{"Ages"} & & \underbrace{\text{"18 \quad - \quad 55 \quad years \quad old"}} \\ \downarrow & & \downarrow \\ \textit{Age} & \xrightarrow[Numeric-Filter]{} & \textit{Eq-Comparison} \end{array}$$

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

$$\begin{array}{cccc} \text{"18"} & \text{"-"} & \text{"55"} & \text{"years"} \\ \downarrow & \downarrow & \downarrow & \downarrow \\ \textit{Eq-Value} & \textit{Eq-Operator} & \textit{Eq-Value} & \textit{Eq-Temporal-Unit} \\ & \textit{[between]} & & \textit{[year]} \end{array}$$

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

$$\textit{Eq-Comparison} \begin{cases} \xrightarrow[Value]{} & \textit{Eq-Value "18"} \\ \xrightarrow[Operator]{} & \textit{Eq-Operator[between]} \text{ "-"} \\ \xrightarrow[Value]{} & \textit{Eq-Value "55"} \\ \xrightarrow[Temporal-Unit]{} & \textit{Eq-Temporal-Unit[year]} \text{ "years"} \end{cases}$$

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.

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 [41] to further normalize the phrase to a value [36, 42]. 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:

$$\begin{tikzcd} \text{"preceding"} \arrow[d] & \text{"six"} \arrow[d] & \text{"months"} \arrow[d] \\ \begin{array}{c}\textit{Eq-Temporal-Period} \\ \textit{[past]}\end{array} & \textit{Eq-Value} & \begin{array}{c}\textit{Eq-Temporal-Unit} \\ \textit{[month]}\end{array} \end{tikzcd}$$

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.

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:

$$\begin{tikzcd}
\text{"Conditions"} \arrow[d] & \text{"contraindicating"} \arrow[d] & \text{"pioglitazone"} \arrow[d] \\
\mathit{Condition} & \mathit{Contraindication} \arrow[l, "Caused{-}By"'] \arrow[r, "Contraindicates"'] & \mathit{Drug} \\
 & & + \\
 & & \mathit{Drug\text{-}Name}
\end{tikzcd}$$

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.

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 2 shows comparison examples of annotations of the same eligibility criteria using the two corpora in the Brat annotation tool [39].

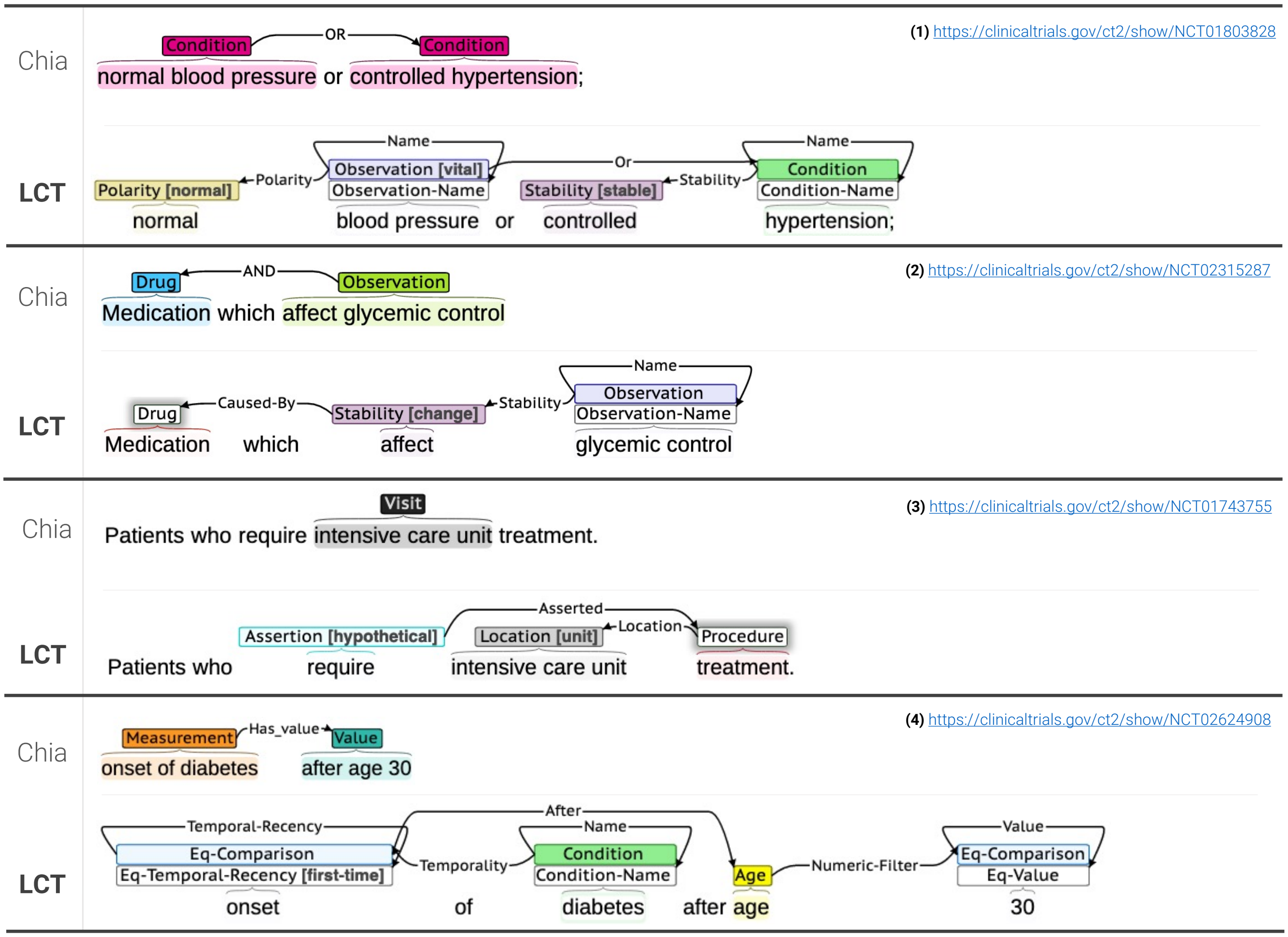

Chia
(1) https://clinicaltrials.gov/ct2/show/NCT01803828
Condition
OR
Condition
normal blood pressure or controlled hypertension;
LCT
Polarity [normal]
Polarity
Name
Observation [vital]
Observation-Name
Or
Stability [stable]
Stability
Name
Condition
Condition-Name
normal blood pressure or controlled hypertension;
Chia
(2) https://clinicaltrials.gov/ct2/show/NCT02315287
Drug
AND
Observation
Medication which affect glycemic control
LCT
Drug
Caused-By
Stability [change]
Stability
Name
Observation
Observation-Name
Medication which affect glycemic control
Chia
(3) https://clinicaltrials.gov/ct2/show/NCT01743755
Visit
Patients who require intensive care unit treatment.
LCT
Assertion [hypothetical]
Asserted
Location [unit]
Location
Procedure
Patients who require intensive care unit treatment.
Chia
(4) https://clinicaltrials.gov/ct2/show/NCT02624908
Measurement
Has_value
Value
onset of diabetes after age 30
LCT
Temporal-Recency
Eq-Comparison
Eq-Temporal-Recency [first-time]
Temporality
After
Name
Condition
Condition-Name
Age
Numeric-Filter
Value
Eq-Comparison
Eq-Value
onset of diabetes after age 30

Figure 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).

4.1.4 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 [39]. 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 [43] 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 1.

Measure	EliIE [5]	Chia [4]	**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 1: Annotation statistics for EliIE, Chia, and LCT corpora.

4.1.5 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.

4.1.6 Baseline prediction

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 [44] neural architectures. For BERT-based prediction, we used two pretrained models trained on published medical texts, SciBERT [45] and PubMedBERT [46]. 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 [47], also using the pretrained SciBERT model. Table 2 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 [48]
Relation Extraction	BERT & R-BERT	Pretrained Model	SciBert
		Learning Rate	0.00003

Table 2: 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 [2]. Results of representative entities and relations are shown in Tables 3 and 4.

4.1.7 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 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. These results indicate relative uniformity and consistency in the corpus across both single- and double-annotated 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 5.

[2] 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: 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**
Alt. 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
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 4: Baseline relation prediction scores (%, Precision / Recall / $\mathbf{F_1}$). 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.*

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 5: 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.

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.

4.1.8 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 not a significant issue.

4.1.9 Conclusion

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.

4.2 Aim 2: Query Generation Methods Development and Evaluation

In Aim 2 we focus on development and evaluation of methods for query generation and reasoning on eligibility criteria. We divide this aim into two sub-aims: (1) Development of a gold standard eligibility criteria logical forms corpus, and (2) Creation of methods for query generation utilizing the corpus in (1). This aim is largely complete and currently in the analysis phase.

4.2.1 Related Work

Various methods for matching eligibility criteria to cohorts of patients using NLP have been put forth by the research community [42, 49, 50, 33, 51, 52, 53, 54, 55]. 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** to 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 criteria.

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.

Next we briefly describe recent relevant work in each category.

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 [56]. Yuan *et al* developed Criteria2Query, a hybrid information extraction (IE) pipeline and application which uses both rules and machine learning to generate database queries on an OMOP database. This work was expanded by Fang *et al*, who added functionality for iterative query generation via human correction and adjustment [50]. 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 [57, 58, 59, 60, 53]. Park *et al* [58] experimented with transforming medical questions generated in the MIMICSQL data set [61, 59] 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* [60] 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* [51] used hybrid rule-based heuristics and sentence pattern-matching to detect criteria structure, 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 [49] utilized the BERT Transformer architecture [44] and Lucene [62] 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* [53] 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* [33] 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 [52], 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 [63] and normalization techniques to infer cancer staging. Building upon earlier work [64, 65, 66], Baader *et al* [67] 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) ∧ Diabetes_mellitus_type_1(y)". Liu *et al* [54] 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 "#Inclusion features['StartDate'] >= demographics['BirthDate'] + @YEARS(18)". Parsed eligibility criteria were then executed on a proprietary database schema to determine eligible patients.

4.2.2 Sub-Aim 1: Creation of a Criteria to Logical Form Gold Standard

Query generation for LeafAI was originally envisioned as a system which generates queries using rules executed upon predicted named entities and relations in the form of graphs. In practice, we found this system to be difficult to manage given the need for ever more rules, as well as error-prone in handling new unseen criteria, which in turn required the creation of additional rules.

As a solution to this, we instead explored the transformation of eligibility criteria into intermediate "logical form" representations, then generating SQL statements from the parsed logical forms. Intermediate representations (IRs) have a long history in computer science and NLP. IRs remove "noise" unnecessary to a given task and more closely represent underlying semantics while being agnostic to particulars of a final executed form (see Herzig *et al* [68] for an examination of IR-based SQL generation approaches). In related work, Roberts and Demner-Fushman [69] proposed an IR 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 highly generalizable, but also difficult to translate directly into SQL statements as multiple predicates (e.g., *has_problem* and *time_within*) may correspond to a variable number of SQL statements, depending on context.

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

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

We named the corpus produced from this sub-aim the Leaf Logical Forms (LLF) corpus. We developed annotation guidelines for the LLF corpus using a simplification of entities and relations from the preceding LCT corpus discussed in Aim 1. Generally speaking, LCT *entities* correspond 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.

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 3.

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.

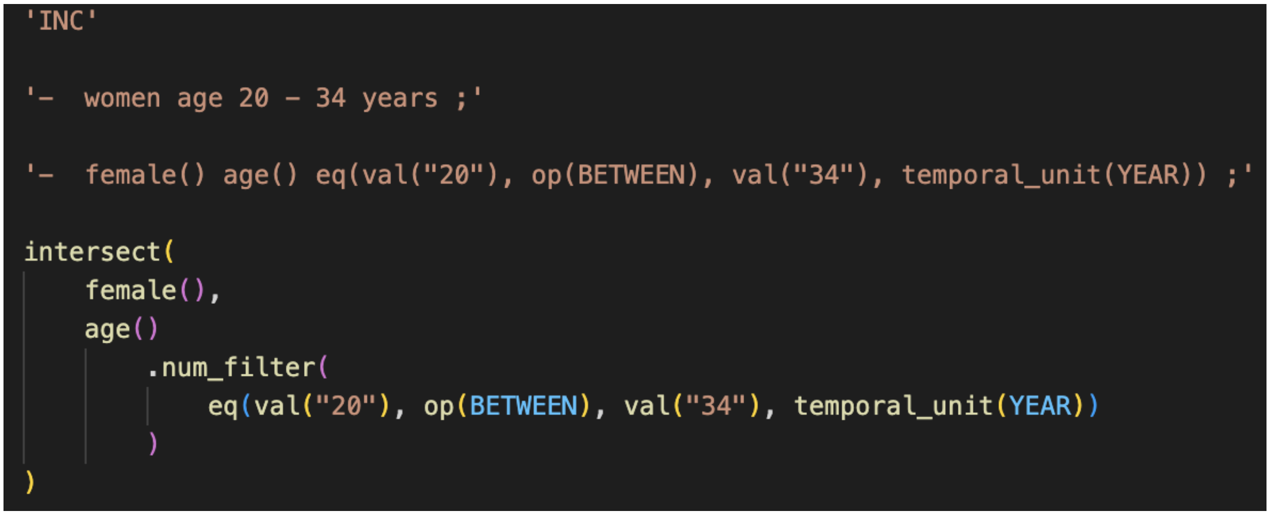
After annotations were completed, we experimented with predicting logical forms by fine-tuning T5 [70] Seq2Seq models. The T5 architecture and pre-trained models are widely used for and achieve at or near state-of-the-art for many machine translation and semantic parsing tasks.

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* [71] 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* 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.

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 "Standard" style. Example inputs, outputs, and training results are shown in Table 6.



```
'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 3: A 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.

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

We found that our Standard logical forms achieved the highest performance using both BLEU [72] and ROUGE-L [73] scores, two commonly used metrics in measuring Seq2Seq performance. Replacing raw tokens with function names corresponding to named entities also significantly improved performance (+14.7%, comparing raw text to Standard input styles), demonstrating that leveraging the LCT corpus to generate augmented text achieved relatively high performance ($> 93\%$ BLEU score) for this task. 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 our work in sub-aim 2.

4.2.3 Sub-Aim 2: Query Generation

In sub-aim 2, we developed the LeafAI query engine, an application capable of generating database queries for cohort discovery from free-text eligibility descriptions. This sub-aim contributes the following:

1. A novel database schema annotation and mapping method to enable data model-agnostic query generation from natural language.

2. Methods for transforming and leveraging intermediate logical representations of eligibility criteria.

3. Methods for dynamically reasoning upon non-specific criteria using an integrated knowledge base of biomedical concepts.

4.2.4 System Architecture

The LeafAI query engine was designed using a modular, micro service-based architecture with a central API (Application Program Interface) which orchestrates end-to-end query generation. Inter-module communication is performed using gRPC [74], 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. A diagram of the LeafAI query engine architecture is shown in Figure 4.

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.

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 named entities. Any entities found with a hypothetical *Assertion* relation (e.g., "could become pregnant") are excluded.

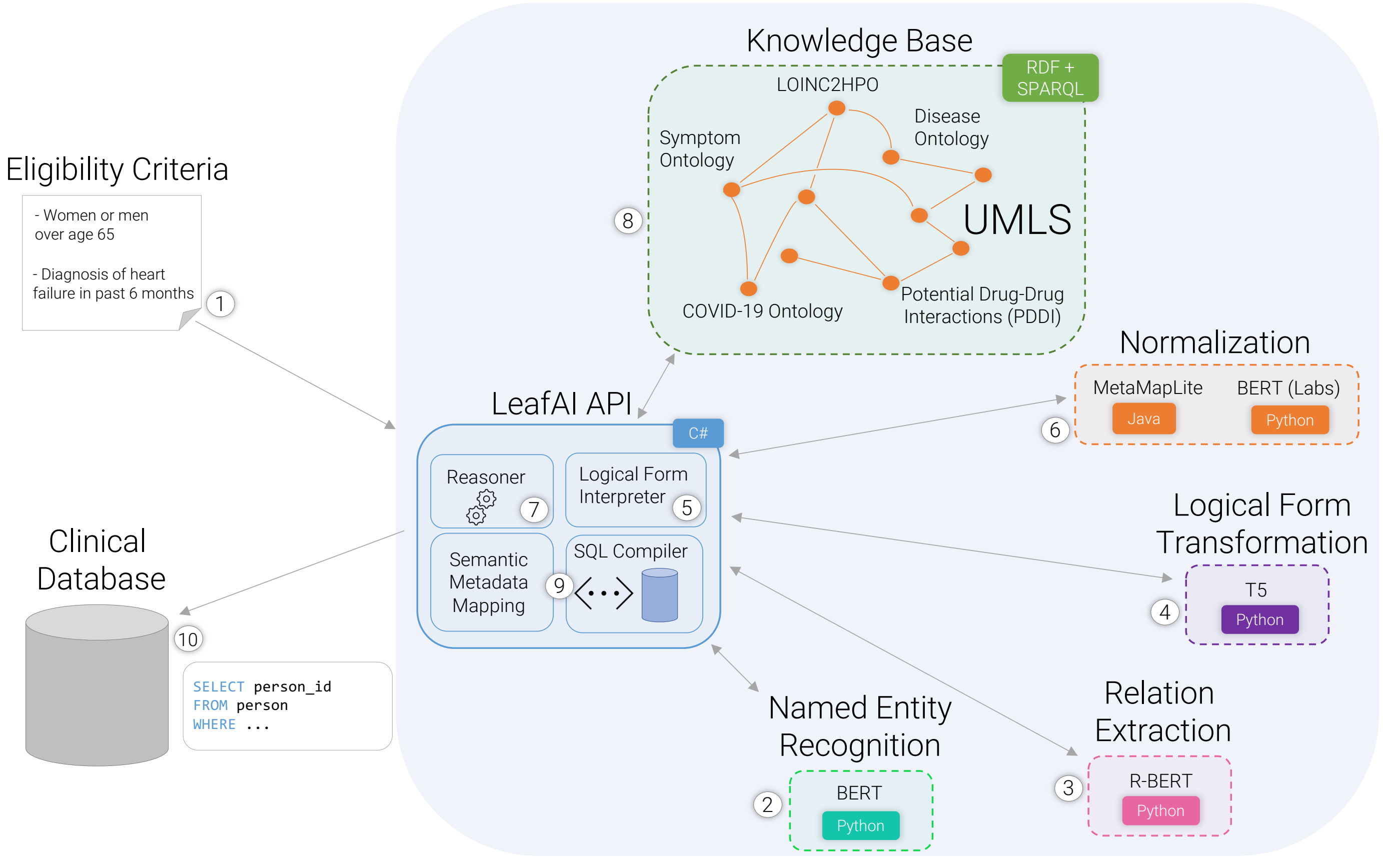
4. The input eligibility criteria are transformed by replacing spans of "raw" text with logical form names as in sub-aim 1. The resulting augmented criteria are inputted into our fine-tuned T5 model, which outputs a predicted logical form string.

5. A logical form interpreter module implemented as a recursive descent parser [76] reads the logical form string 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 *lab("hemoglobin A1c")*) are normalized into one or more corresponding UMLS concepts.

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 knowledge base (KB), concept by concept.



- Women or men over age 65

- Diagnosis of heart failure in past 6 months

```
SELECT person_id
FROM person
WHERE ...
```

Figure 4: LeafAI query architecture. Inter-module communication is performed using the gRPC framework. Individual modules are deployed as Docker [75] containers and communicate solely with the central API, which orchestrates query generation and handles query generation requests.

9. The 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).

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.

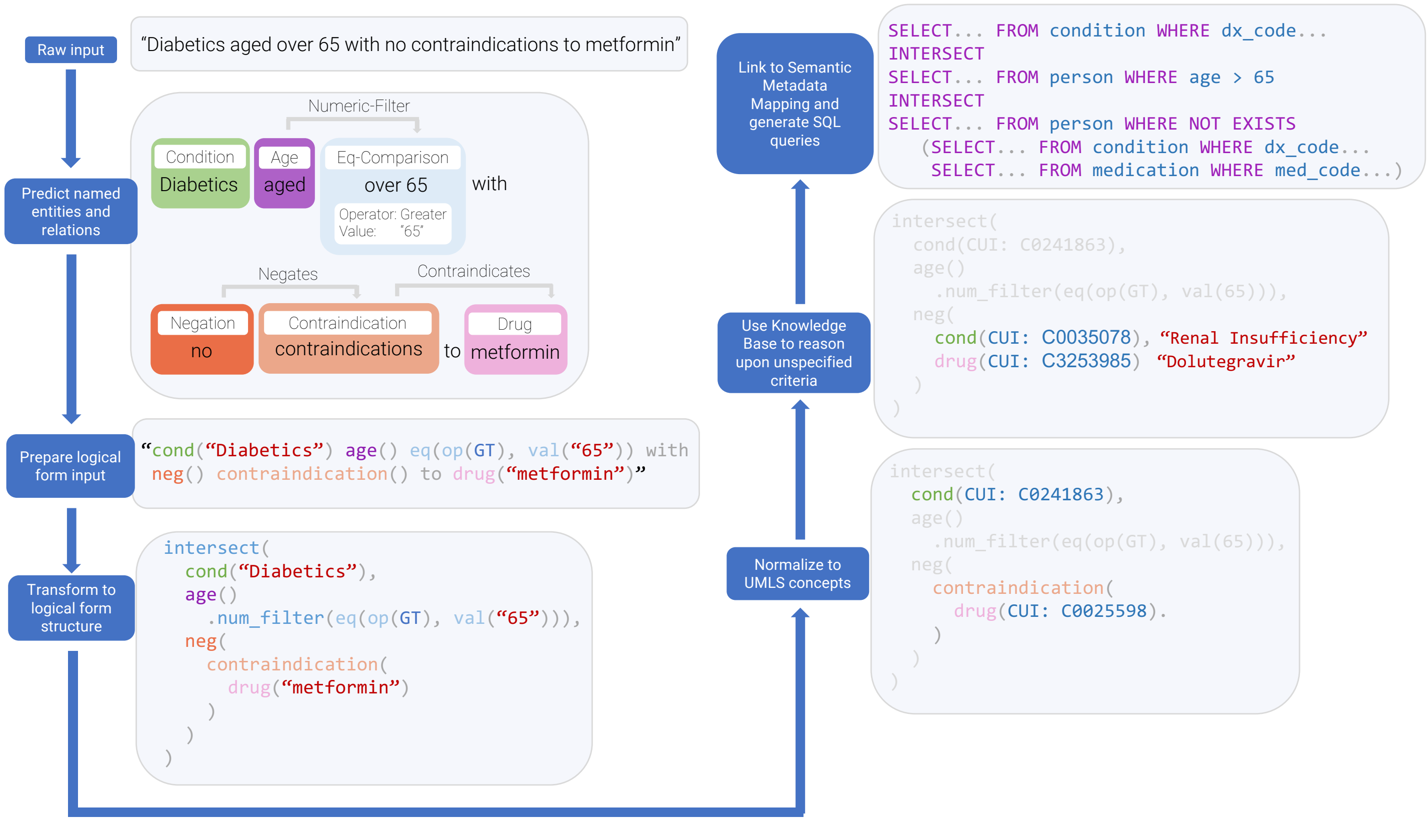
Figure 5 illustrates an example of this process. Next we examine these steps in detail.

Named entity recognition and relation extraction

We use the Leaf Clinical Trials (LCT) corpus [77] to train two BERT-based [44] NER extractors, one each for LCT general- and fine-grained-entities. Next, we perform relation extraction by enumerating named entity pairs similarly using a BERT-based model also trained on the LCT corpus.

Logical form transformation

As described in Sub-Aim 1, we leverage a fine-tuned T5 model for predicting logical forms. As inputs to the T5 model we use augmented text generated using predicted named entities. After prediction, the output logical form string is then instantiated into an AST of nested in-memory logical form objects using a recursive descent parser within the API.


Raw input
"Diabetics aged over 65 with no contraindications to metformin"
Predict named entities and relations
Numeric-Filter
Condition
Diabetics
Age
aged
Eq-Comparison
over 65
Operator: Greater
Value: "65"
with
Negates
Contraindicates
Negation
no
Contraindication
contraindications
to
Drug
metformin
Prepare logical form input
Transform to logical form structure
Normalize to UMLS concepts
Use Knowledge Base to reason upon unspecified criteria
Link to Semantic Metadata Mapping and generate SQL queries

```
"cond("Diabetics") age() eq(op(GT), val("65")) with
neg() contraindication() to drug("metformin")"
```

```
intersect(
  cond("Diabetics"),
  age()
    .num_filter(eq(op(GT), val("65"))),
  neg(
    contraindication(
      drug("metformin")
    )
  )
)
```

```
intersect(
  cond(CUI: C0241863),
  age()
    .num_filter(eq(op(GT), val(65))),
  neg(
    contraindication(
      drug(CUI: C0025598).
    )
  )
)
```

```
intersect(
  cond(CUI: C0241863),
  age()
    .num_filter(eq(op(GT), val(65))),
  neg(
    cond(CUI: C0035078), "Renal Insufficiency"
    drug(CUI: C3253985)  "Dolutegravir"
  )
)
```

```
SELECT... FROM condition WHERE dx_code...
INTERSECT
SELECT... FROM person WHERE age > 65
INTERSECT
SELECT... FROM person WHERE NOT EXISTS
   (SELECT... FROM condition WHERE dx_code...
    SELECT... FROM medication WHERE med_code...)
```

Figure 5: LeafAI query generation processes

Concept normalization

We consider a logical form "named" if it contains a free-text value surrounding by quotes. For logical forms besides laboratory values, we used MetaMapLite [78, 79]. Normalization using MetaMapLite can often result in high recall but low precision, as MetaMapLite matches potential candidate concepts across the UMLS. To improve normalization precision, we employ two strategies. First, we compare term frequency-inverse document frequency (tf-idf) on MetaMapLite predictions, dropping UMLS concepts whose 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. Using our tf-idf strategy removes the erroneous infection concepts. Next, our NER component also us to further improve precision by filtering the predicted UMLS concepts to only those of specific semantic types. For example, we limit condition concepts to only those which include semantic types of signs or symptoms, diseases or syndromes, and so on.

Laboratory values also present a particular challenge, as LeafAI requires 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. For example, a search for "platelet count" with MetaMapLite returns the concept "Platelet Count Measurement" (C0032181), but not the needed concept of "Platelet # Bld Auto" (C0362994). Thus similar to Lee and Uzuner with medications [80], we trained a BERT model specifically for normalization of lab tests.

Reasoning using an integrated knowledge base

For reasoning and derivation of ICD-10, LOINC, and other codes for UMLS concepts, we designed a knowledge base (KB) accessible via SPARQL queries and stored as RDF triples. The core of our KB is the UMLS, derived using a variation of techniques created for ontologies in BioPortal [81]. To further augment the

UMLS, we mapped and integrated the Disease Ontology [82], Symptom Ontology [83], COVID-19 Ontology [84], Potential Drug-Drug Interactions [85], LOINC2HPO [86], and the Disease-Symptom Knowledge Base [87]. We then developed SPARQL queries parameterized by UMLS concepts for various scenarios which leveraged our KB, such as contraindications to treatments, symptoms of diseases, and so on. Using LOINC2HPO mappings further allows us to infer phenotypes by lab test results rather than diagnosis codes alone. For example, given the logical form *cond("Hypercalcemia")*, our system will search for abnormally high lab results of calcium [mass/volume] (LOINC: 17861-6) in addition to diagnosis codes.

Together our KB, nested logical forms, and inside-to-outside normalization and reasoning 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.

Query generation using semantic metadata mapping

To enable database schema-agnostic query generation, we leverage a subset of codes within the UMLS in what we define as a semantic metadata mapping, or SMM. An SMM is described using JSON, and includes a listing of available databases, tables, columns, and so on. Critically, these database artifacts are "tagged" using UMLS concepts. An example of this can be seen in Figure 6, which shows strategies by which a given criterion can be used to generate schema-specific queries by leveraging different SMMs. In cases where the LeafAI query engine finds more than one means of querying a concept (e.g., two SQL tables for diagnosis codes), the queries are combined in a UNION statement.

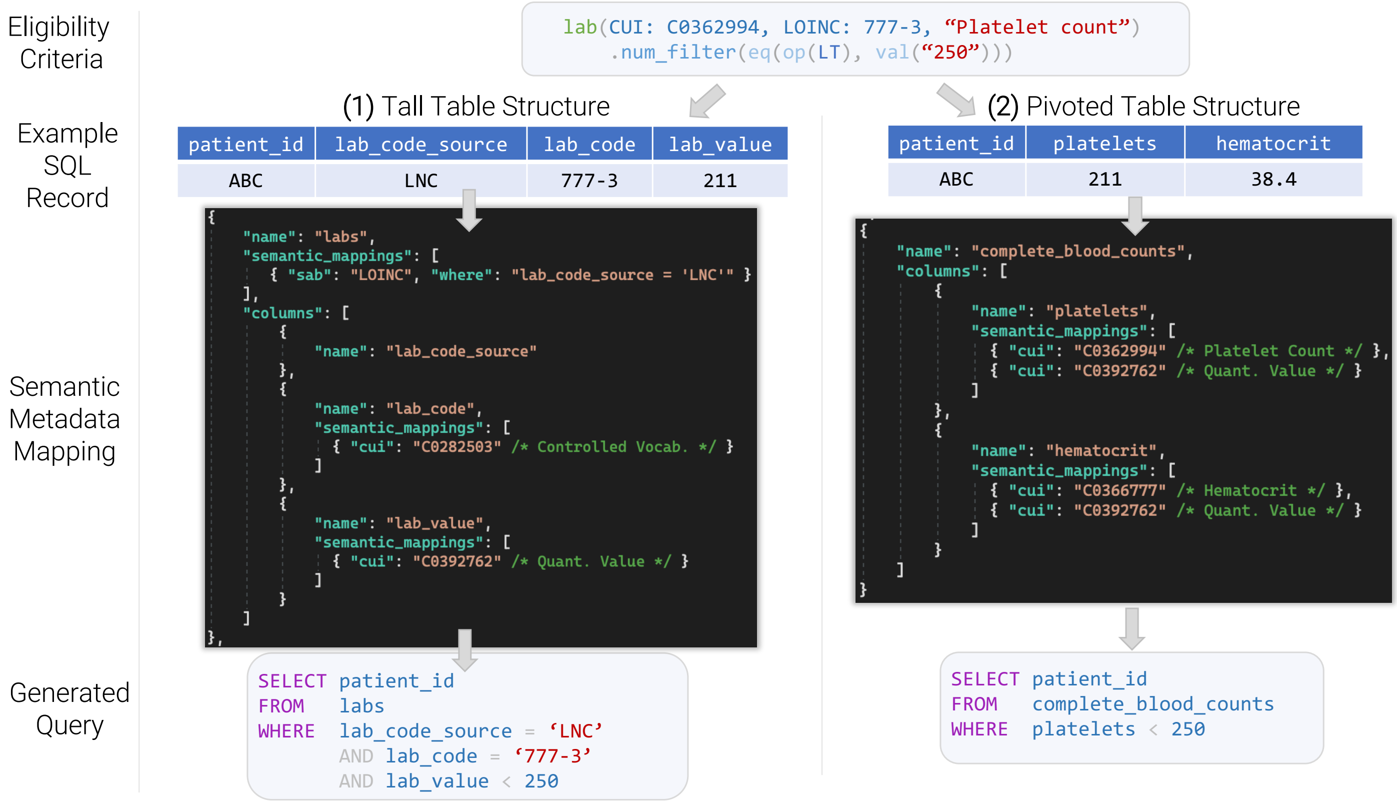
4.2.5 Evaluation

An NLP-based system for finding patients based on eligibility criteria should be reasonably expected to find many or most patients enrolled in a real clinical trial - with the assumption that patients enrolled in said trial correctly met the necessary criteria as determined by study investigators. While there are caveats to this approach (for example, certain structured data may be missing for some patients), we aimed to establish a new baseline by which tools such as ours are evaluated in their ability to handle real-world eligibility criteria and clinical data.

For comparison, we are analyzing database queries created by a human database programmer experienced with clinical databases and data extraction. Our evaluation is being performed as follows:

1. We extracted metadata on 165 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 within the eligibility criteria (besides the phrases "Inclusion Criteria" and "Exclusion Criteria") were less than or equal to 30. We excluded 41 trials with multiple sub-groups, as it would not be possible to know which eligibility criteria applied to which sub-group of enrolled patients.

2. Using the "condition" field for each trial within metadata from `https://clinicaltrials.gov`, we filtered and grouped the remaining 124 trials into only those related to predetermined groups: Cardiology, COVID-19, Crohn's Disease, Multiple Sclerosis, Diabetes Mellitus, Hepatitis C, and Oncology.



```
lab(CUI: C0362994, LOINC: 777-3, “Platelet count”)
    .num_filter(eq(op(LT), val(“250”)))
```

(1) Tall Table Structure

patient_id	lab_code_source	lab_code	lab_value
ABC	LNC	777-3	211

```
{
    "name": "labs",
    "semantic_mappings": [
        { "sab": "LOINC", "where": "lab_code_source = 'LNC'" }
    ],
    "columns": [
        {
            "name": "lab_code_source"
        },
        {
            "name": "lab_code",
            "semantic_mappings": [
              { "cui": "C0282503" /* Controlled Vocab. */ }
            ]
        },
        {
            "name": "lab_value",
            "semantic_mappings": [
              { "cui": "C0392762" /* Quant. Value */ }
            ]
        }
    ]
},
```

```
SELECT patient_id
FROM   labs
WHERE  lab_code_source = ‘LNC’
       AND lab_code = ‘777-3’
       AND lab_value < 250
```

(2) Pivoted Table Structure

patient_id	platelets	hematocrit
ABC	211	38.4

```
{
    "name": "complete_blood_counts",
    "columns": [
        {
            "name": "platelets",
            "semantic_mappings": [
              { "cui": "C0362994" /* Platelet Count */ },
              { "cui": "C0392762" /* Quant. Value */ }
            ]
        },
        {
            "name": "hematocrit",
            "semantic_mappings": [
              { "cui": "C0366777" /* Hematocrit */ },
              { "cui": "C0392762" /* Quant. Value */ }
            ]
        }
    ]
}
```

```
SELECT patient_id
FROM   complete_blood_counts
WHERE  platelets < 250
```

Figure 6: The LeafAI query engine's SQL query generation process using two hypothetical database schema to generate queries for platelet counts (shown in logical form after normalization). This example illustrates the flexibility of LeafAI's semantic metadata mapping system 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. The LeafAI 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.

3. We randomly chose 1 trial from each group, with the exception of Oncology, where we chose 2 trials.

4. The human programmer was provided the eligibility criteria for each of the 8 selected trials and instructed to (1) ignore criteria which cannot be computed, (2) make a reasonable effort to reason upon non-specific criteria (e.g., symptoms for a condition), (3) not check whether patients found by a query enrolled within a trial, and (4) skip criteria which cause an overall query to find no eligible patients, as they typically indicate missing data.

5. We generated queries using the LeafAI query engine, modifying the generated SQL to output the number of matched patients at each step.

6. In order to ensure results returned would be limited to only data available during the time of each trial, for each system we (1) replaced references to the SQL function for generating a current timestamp 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 the end of a trial.

4.2.6 Results

Current results of our experiments are shown in Table 7.

				LeafAI		Human	
Condition	**ID**	**Lines of Criteria**	**Enrolled**	**Matched**	**Eligible**	**Matched**	**Eligible**
CLL	NCT04852822	4	83	80 (96%)	3,252		
Hepatitis C	NCT02786537	8	42	33 (78%)	9,529	33 (78%)	4,981
Crohn's Disease	NCT03782376	9	16	0 (0%)	113		
Cardiac Arrest	NCT04217551	12	27	12 (44%)	4,792		
COVID-19	NCT04501952	13	41	0 (0%)	0		
Multiple Sclerosis	NCT03621761	14	196	149 (76%)	5,674		
Type 1 Diabetes	NCT03335371	18	11	0 (0%)	1,006		
Ovarian Cancer	NCT03029611	25	11	10 (91%)	1,667		
Mean				48.2%		78.5%	
Total		103	427	284 (66%)	27,225	33 (78%)	4,981

Table 7: 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. The "# Crit." column refers to the number of lines of eligibility criteria which were not empty and did not contain the phrases "Inclusion Criteria" or "Exclusion Criteria".

While our analysis of the queries from each system is ongoing, we found that a total of 284 of the 427 (66%) patients enrolled across the 8 trials were successfully matched by LeafAI, of a total of 27,225 patients predicted to be eligible. LeafAI found over 40% of enrolled patients in 5 of 8 (62%) of trials and zero patients in the remaining 3 trials. Figure 7 shows matched patients from each trial's executed queries tracked step-by-step.

While LeafAI executed queries for Crohn's Disease (NCT03782376) and Type 1 Diabetes Mellitus (NCT03335371) ultimately matched zero patients, as Figure 7 shows, initially in both trials LeafAI matched at least have of the enrolled patients, who were subsequently were found ineligible in later criteria. Preliminary error analysis also reveals certain interesting findings. For example, line 12 of trial NCT02786537 for Hepatitis C excludes "Child Pugh (CTP) B or C Cirrhosis)", while 11 of the 42 enrolled patients had diagnosis codes for cirrhosis, and were thus excluded by LeafAI.

4.2.7 Limitations and Future Work

To the best of our knowledge, the query generation and associated methods of LeafAI are the current state-of-the-art for the task of cohort discovery using a natural language interface. Nevertheless our system has a number of limitations. First, while we evaluated our query generation methods using a random sample of 8 actual clinical trials, those trials are a small subset of trials performed at the University of Washington and thus performance may vary significantly if our evaluation was expanded to a greater number of trials and disease domains. Second, while capable of reasoning across a large number of different scenarios and diseases, our KB and reasoning module use rules and pre-determined queries which likely fall short and fail to capture results in certain cases. Moreover, our reasoning module is incapable of capturing the nuance and complexities between many clinical concepts. For example, in Figure 5, renal insufficiency and Dolutegravir are shown as contraindications to Metformin, but in reality for many patients those may not

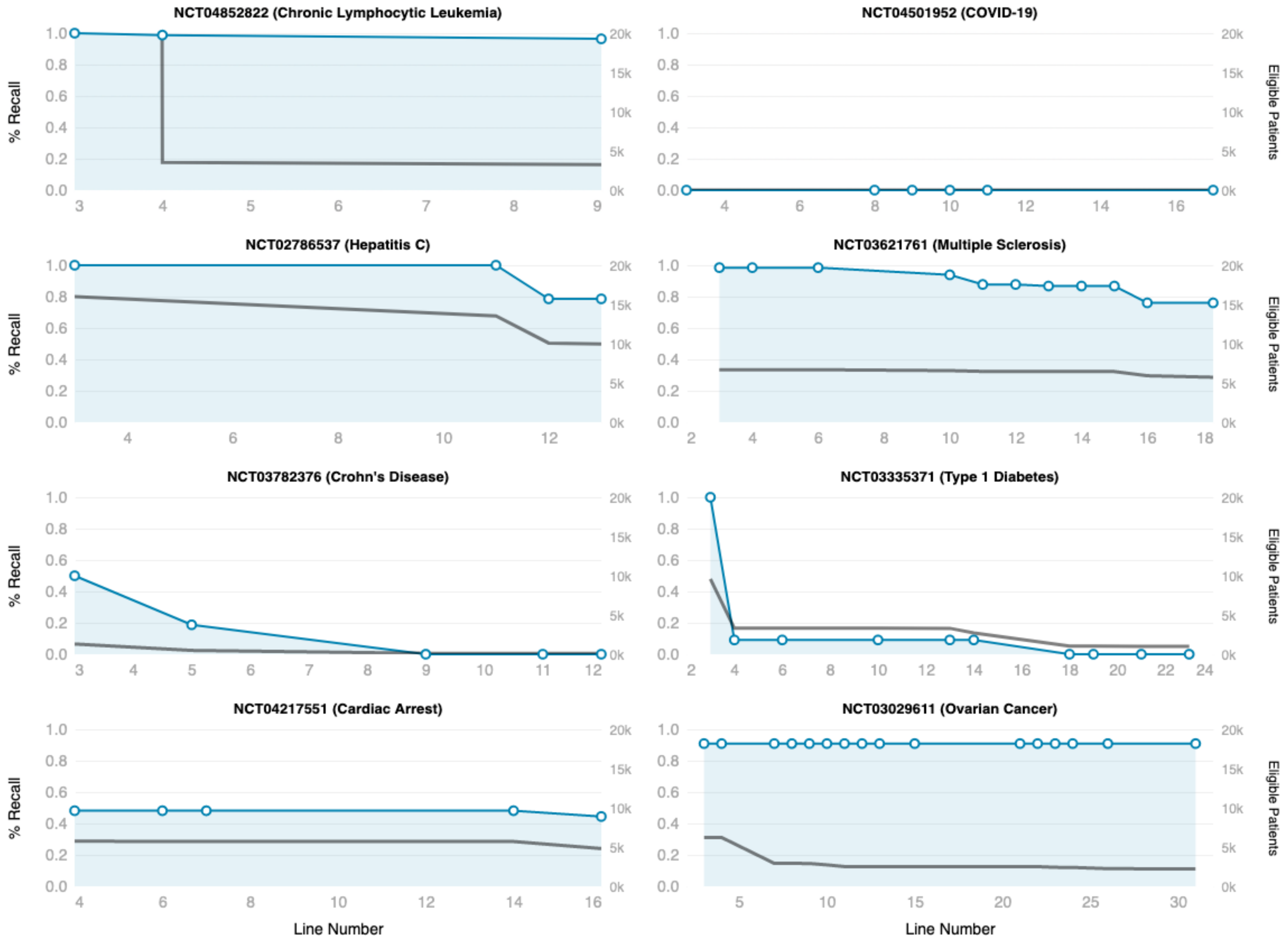


Figure 7: Longitudinal results listing results of patients found at each step in the query process for each trial. The blue line indicates % Recall (left-most Y axis) while the gray line indicates the number of eligible patients found (right-most Y axis). The X axis represents the line number within the free-text eligibility criteria. Dots indicate that the LeafAI query engine executed an eligibility criterion query.

be absolute contraindications, depending on other health factors. We intend to address this by allowing users to add or remove LeafAI KB-derived concepts which they don't wish to include in Aim 3 (e.g., a user could choose to keep diagnoses of renal insufficiency while ignoring prescriptions of Dolutegravir). Last, while our logical forms representation is uniquely flexible and demonstrably capable of representing many eligibility criteria, there are nonetheless likely cases where our representation does not fully capture the semantics and nuance intended within certain criteria.

In future work, we intend to explore extending our logical form representation for general-purpose question-answering. Leveraging the outer-most functions introduced by Roberts and Demner-Fushman such as *latest()*, λ*()*, and *min()*, our logical forms could be "wrapped" such that our logical forms are used for data retrieval while Roberts and Demner-Fushman's functions are executed to analyze returned data to answer a question. For example, the question

"*Did the patient's temperature exceed 38C in the last 48 hrs?*"

could be represented as

```
λ(
    measurement(”temperature”)
        .num_filter(eq(op(GT), val(”38”), unit(”C”)))
        .temporality(eq(op(LTEQ), val(”48”), unit(HOUR)))
)
```

where the *measurement()* function serves as a logical form for retrieval of relevant temperature data, while the *λ()* function is executed to analyze the output and return a Boolean yes/no answer.

While general-purpose question-answering is not a specific aim of this project, the methods we have developed have potential to be a foundation for exciting future research directions.

4.2.8 Conclusion

We created the LLF corpus, a gold-standard human-annotated corpus of clinical trial eligibility criteria and corresponding logical forms. The Seq2Seq model fine-tuned on this corpus achieved a relatively high BLEU score of 93.5%. Using models trained on the LCT and LFF corpora as well as an integrated KB and other modules, we successfully developed a state-of-the-art query generation approach potentially capable of rivaling that of a human programmer in certain cases, though our analysis is ongoing. Our methods in this aim enable the web application developed in Aim 3.

4.3 Aim 3: Web Application Development and Evaluation

Aim 3 will focus on the development of an interactive web application, LeafAI, capable of generating and executing user queries.

4.3.1 Related Work

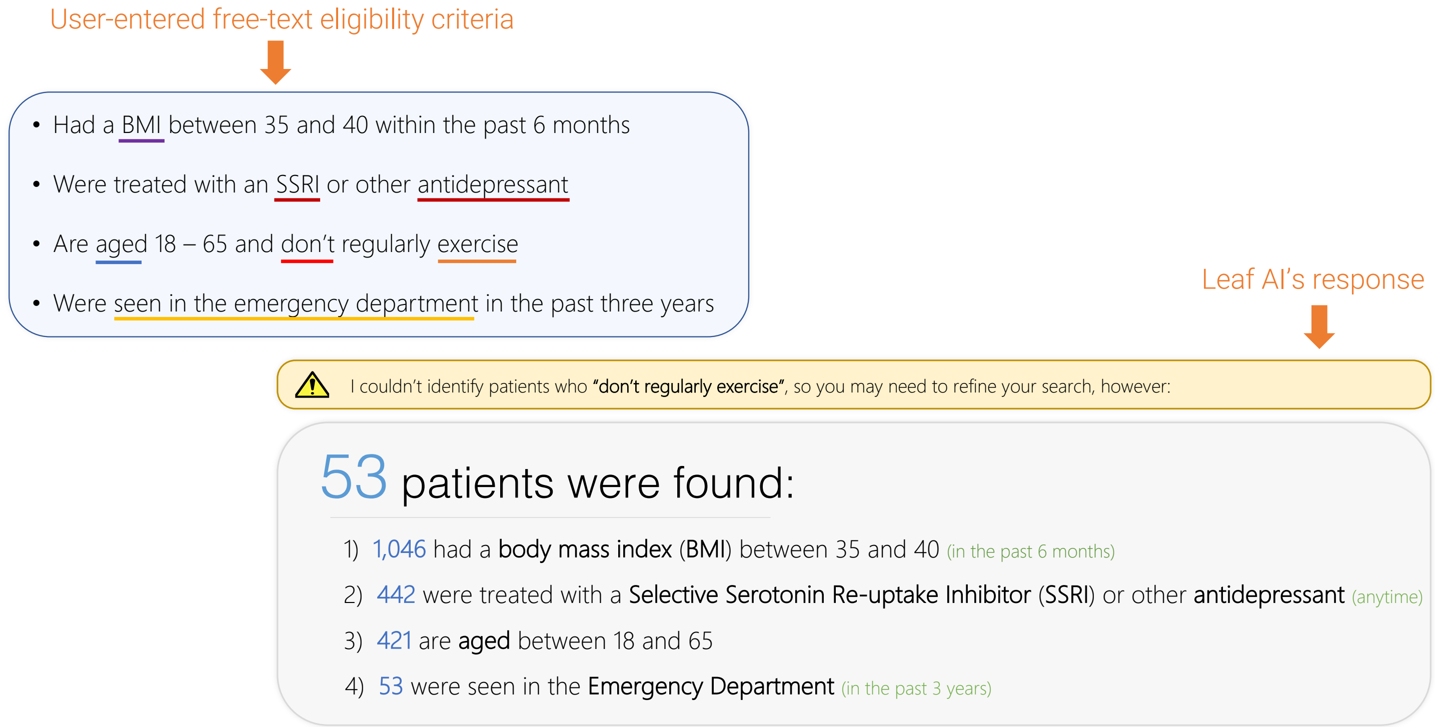
Experimental systems enabling the querying of databases using natural language interfaces have been in development since the 1960s. Using relatively simple rule-based parsing systems, Woods [88] created a system for asking natural language questions of a moon rock database, while Epstein and Walker [89] similarly designed a natural language interface for a melanoma database. Decades later, Katz *et al* created START [90], 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 [91], 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?". 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 [42, 50]. Criteria2Query offers a web-based simple and friendly user interface for inputting free-text eligibility criteria. The system analyzes user inputs and returns a highlighted listing of named entities it identified, as well as a potential list of patients meeting those criteria from an OMOP database.

4.3.2 Application Design

LeafAI will be developed using a 3-tier architecture similar to Leaf [2]: a back-end with clinical and application databases, and server hosting an API (discussed in Aim 2), and a front-end web application. The key innovation of the web application will be its user interface, a chat-like design we expect will be familiar to most users of applications such as Microsoft Teams or Slack.

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 proposed chat interface is shown in Figure 8. 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.



User-entered free-text eligibility criteria

- Had a BMI between 35 and 40 within the past 6 months
- Were treated with an SSRI or other antidepressant
- Are aged 18 – 65 and don't regularly exercise
- Were seen in the emergency department in the past three years

Leaf AI's response

⚠ I couldn't identify patients who **"don't regularly exercise"**, so you may need to refine your search, however:

53 patients were found:

1) 1,046 had a **body mass index** (BMI) between 35 and 40 (in the past 6 months)
2) 442 were treated with a **Selective Serotonin Re-uptake Inhibitor** (SSRI) or other **antidepressant** (anytime)
3) 421 are **aged** between 18 and 65
4) 53 were seen in the **Emergency Department** (in the past 3 years)

Figure 8: A example mockup of the LeafAI user interface. User-entered criteria are shown in the above left, while LeafAI responses are shown in the lower-right. User criteria are displayed and executed in order, with each count of patients representing a subset of the preceding count.

It is important to emphasize that LeafAI will respond using a visual form that is chat-*like*, rather than itself being a chat*bot* (i.e., a programmatic conversation agent). While the user interface and query generation methods described in Aim 2 lay a potential foundation for general-purpose question-answering more akin to a conversation agent, we leave that to future work and limit our scope in this project to cohort discovery. The user interface of LeafAI will have the following goals:

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

2. **Rapid feedback and explainability**. LeafAI while perform NER and normalization as users are typing, similar to prefetching search engine responses. This will allow users to preemptively detect concepts and queries which may return unexpected results. LeafAI will also display incremental query results in real-time, providing users faster feedback so users may avoid waiting until all query results are complete.

3. **Direct editing of responses enabling iteration**. As LeafAI 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.

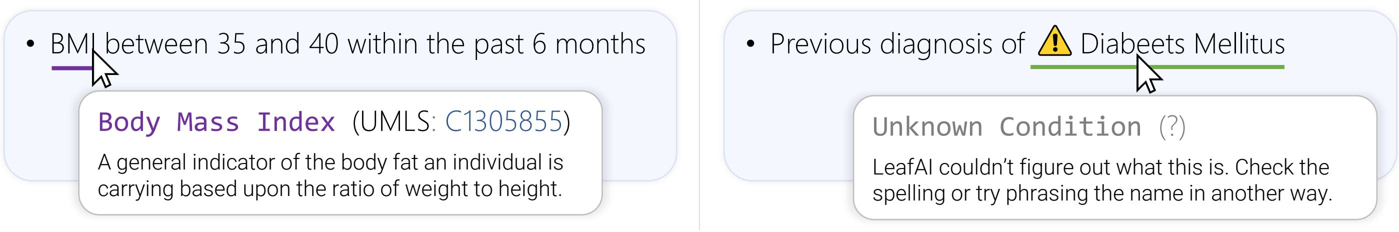
Accessible History

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" [92]. Put another way, user goals and perceptions may change over the course of their 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" [93], 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 [94, 93].

Rapid Feedback and explainability

Users' sense of system latency and responsiveness can significantly affect their satisfaction in using a tool [95, 96, 97]. 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 [98, 99]. LeafAI will employ 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.

First, consider a hypothetical case where a user begins to type a criteria but misspells "Diabetes Mellitus". The user may take seconds or even minutes of additional typing while adding new criteria without knowing of 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. We aim to avoid this scenario by preemptively performing NER and normalization while users are typing. Examples of this are shown in Figure 9. Named entities found within user criteria will be underlined and interactive, enabling users to better estimate whether their queries will succeed or not.



- BMI between 35 and 40 within the past 6 months

Body Mass Index (UMLS: C1305855)

A general indicator of the body fat an individual is carrying based upon the ratio of weight to height.

- Previous diagnosis of ⚠ Diabeets Mellitus

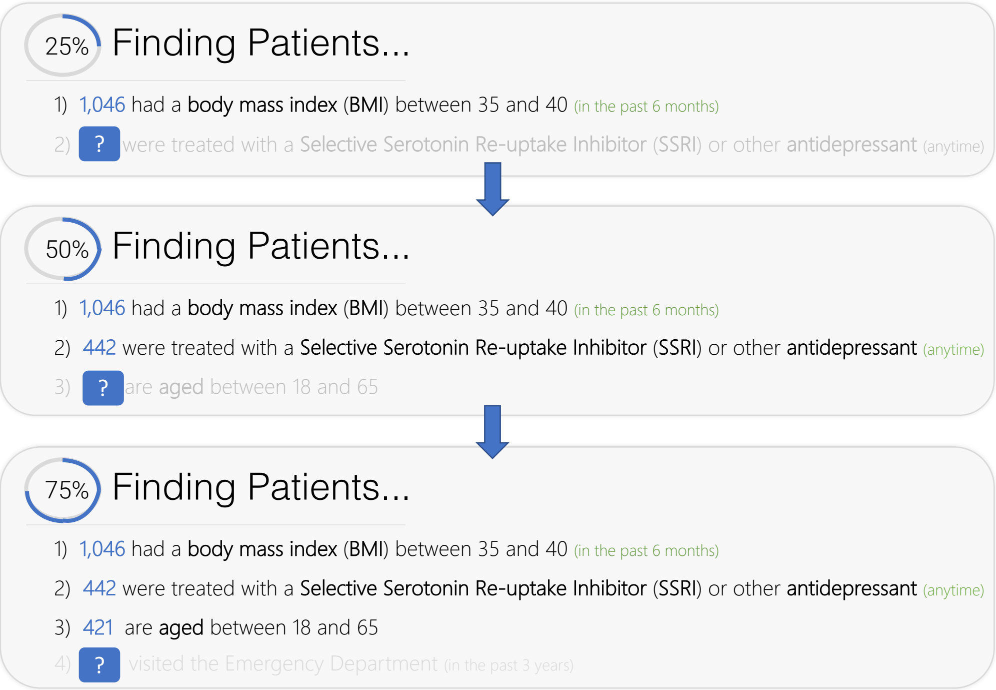
Unknown Condition (?)

LeafAI couldn't figure out what this is. Check the spelling or try phrasing the name in another way.

Figure 9: 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 will show results as they are reported using a streaming interface. As a result, users will not need to wait until all queries are complete (as in tools such as Leaf and i2b2). An example of this is shown in Figure 10.

Taken together, these features for rapid feedback and transparency also enable *explainability* of system actions. That is, rather than simply returning a final count of patients meeting criteria, LeafAI will provide 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.). We currently plan for system language for explaining query results to be generated by templated English expressions mapped to logical form types "slot-filled" using normalized UMLS concept names. By avoiding being a "black box", we expect these features to help gain user trust and understanding of both the system's successes and



25% Finding Patients...

1) 1,046 had a body mass index (BMI) between 35 and 40 (in the past 6 months)

2) ? were treated with a Selective Serotonin Re-uptake Inhibitor (SSRI) or other antidepressant (anytime)

50% Finding Patients...

1) 1,046 had a body mass index (BMI) between 35 and 40 (in the past 6 months)

2) 442 were treated with a Selective Serotonin Re-uptake Inhibitor (SSRI) or other antidepressant (anytime)

3) ? are aged between 18 and 65

75% Finding Patients...

1) 1,046 had a body mass index (BMI) between 35 and 40 (in the past 6 months)

2) 442 were treated with a Selective Serotonin Re-uptake Inhibitor (SSRI) or other antidepressant (anytime)

3) 421 are aged between 18 and 65

4) ? visited the Emergency Department (in the past 3 years)

Figure 10: A time-lapse style representation of the user interface asynchronously reporting incremental query results, from top to bottom. This will be performed using two streaming interfaces, one from the clinical database to the API, and a second from the API to the web application.

shortcomings.

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, LeafAI will allow users to immediately and directly edit the responses returned by LeafAI. This workflow is depicted in Figure 11.

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 will be able to simply click and modify the earlier results.

Additionally, as discussed in Aim 2, LeafAI 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, LeafAI will allow users to directly edit reasoned concepts. An example of this is shown in Figure 12.

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.

Model Inference Speed

LeafAI's modules for NER, logical form transformation, lab normalization and so on use large Transformer-based neural models [100] 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, we intend to quantize our models, a process for converting 32-bit floating point values within model weights and biases to 8-bit integers. Quantization

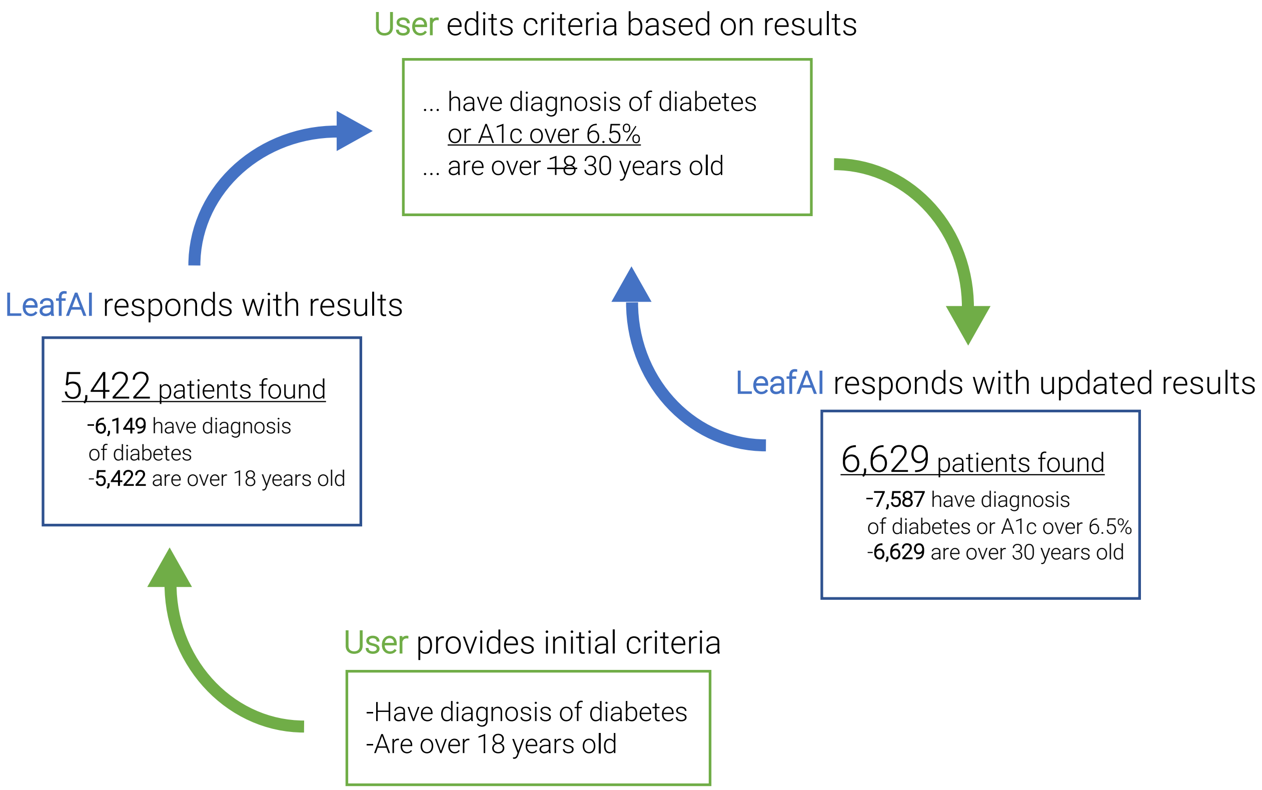
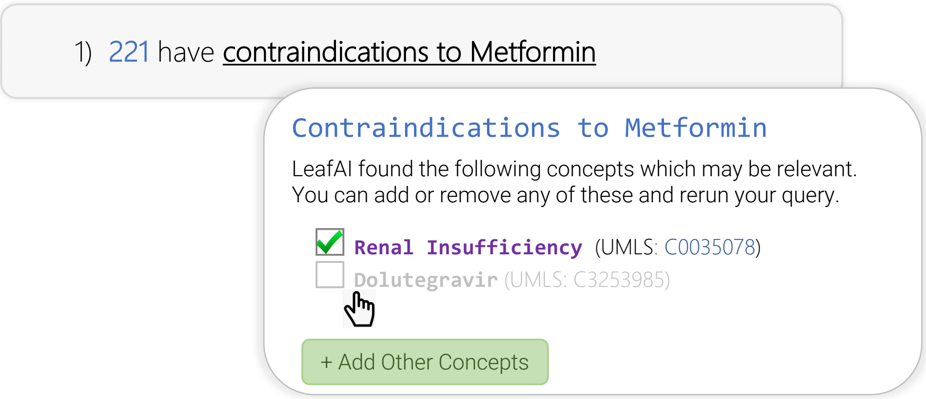

User edits criteria based on results
... have diagnosis of diabetes
or A1c over 6.5%
... are over ~~18~~ 30 years old
LeafAI responds with results
5,422 patients found
-6,149 have diagnosis of diabetes
-5,422 are over 18 years old
LeafAI responds with updated results
6,629 patients found
-7,587 have diagnosis of diabetes or A1c over 6.5%
-6,629 are over 30 years old
User provides initial criteria
-Have diagnosis of diabetes
-Are over 18 years old

Figure 11: An iterative workflow using an example case for adults with Diabetes Mellitus. Users will be able to directly edit results and re-execute their queries while preserving query history, saving user time and preserving previous user actions and findings.



1) 221 have contraindications to Metformin

Contraindications to Metformin

LeafAI found the following concepts which may be relevant. You can add or remove any of these and rerun your query.

☑ Renal Insufficiency (UMLS: C0035078)

□ Dolutegravir (UMLS: C3253985)

Figure 12: An example of a user editing concepts discovered using LeafAI's reasoning system. Users will be able to add or remove reasoned concepts.

has been shown to dramatically reduce model storage size and memory usage and improve inference speed while typically showing limited decreases in performance [101].

4.3.3 Evaluation Metrics

In Aim 2 we demonstrated that LeafAIs query generation methods achieve current state-of-the-art results on the clinical trials we examined. The question remains, however, whether LeafAI's user experience and satisfaction will improve upon traditional drag-and-drop-based cohort discovery tools. Further, it is not clear whether natural language interfaced-based tools such as LeafAI allow users to more accurately find intended cohorts.

To test this, we will evaluate LeafAI in comparison to Leaf in the following controlled experiment:

1. We will enlist 10 researchers who have no experience using Leaf. Half of the researchers (5/10) will be randomly assigned to use Leaf, while the other half will use LeafAI.

2. Both groups will by trained using demonstration tutorials for their respective tool to learn how to run queries and understand results.

3. Groups using Leaf and LeafAI will have access to the same de-identified clinical database[3], which will be composed of patients from 5 randomly selected past clinical trials at UW, as well as 50,000 randomly selected patients not enrolled in the trials. Users will be able to see counts of patients but no patient-level information.

4. Pairs of participants - one from each group - will be tasked to attempt to find patients meeting eligibility criteria for each trial, then save their final query. Each trial will thus have one researcher using Leaf and one researcher using LeafAI to identify patients.

5. After completing their queries, all 10 participants will be asked to complete 2 surveys: the first, Health-ITUES [102], will provide demographic information concerning level of experience in working in clinical research and eligibility screening. The second will be a usability and user satisfaction survey using a 5-point Likert scale.

6. Our analysis will be conducted by comparing (1) user satisfaction with both applications, (2) number of patients matched between generated queries and actual trial enrollments, similar to Aim 2, and (3) length of time spent generating queries, which we will derive from application log files. The Health-ITUES survey will further allow us to group researchers by background and experience level while comparing results.

4.3.4 Limitations

We believe the LeafAI web application will be a significant step forward in cohort discovery in terms of both user experience and query accuracy. However the system will nonetheless have limitations. Within the scope of this project, LeafAI will not include the same features as Leaf, such as the ability to visualize patient demographics or extract row-level data, though we intend to add those in the future. Further, limitations regarding query performance discussed in Aim 2 will also apply in Aim 3, as we expect generated queries to sometimes incorrectly capture eligibility criteria or fail to find the correct patients. LeafAI will also not include a data dictionary or other means of answering possible user questions regarding available clinical data.

4.3.5 Conclusion

In this aim we will create a user-friendly web application enabling iterative state-of-the-art query generation for cohort discovery using user-entered free-text criteria.

[3] We will seek an IRB from the University of Washington Human Subjects Division for permission to extract this data

5 Study Limitations and Future Work

Cohort discovery and clinical data exploration in general is a wide, diverse field, and the methods and applications developed in this project will inevitably not meet certain needs or particularly complex use cases.

The models trained on the LCT and LLF corpora will in some cases predict values incorrectly, both because no model performs perfectly, and also because the gold standard human annotations of those projects are also imperfect and not necessarily representative of all clinical trials. Of the annotators, none were clinical domain experts, which may also impact corpora quality.

Methods for query generation developed in this project also have shortcomings, including cases where normalization or named entity recognition may fail, and the logical form representations we use cannot represent all possible criteria accurately. These flaws inevitably impact certain queries and return potentially incorrect results.

Last, while much of this project deals with the use and identification of data for finding patients, clinical data are tremendously complex and imperfectly representative of reality in terms of human health, recordings of events over time, and so on. Data can be missing, messy, or wrong, and applications such as LeafAI are subject to and impacted by this.

6 Timeline and Summary

We expect this project to be completed in the fall of 2023. Aim 1 is complete, Aim 2 is largely complete, and Aim 3 has already begun. The full timeline is shown in Figure 13.

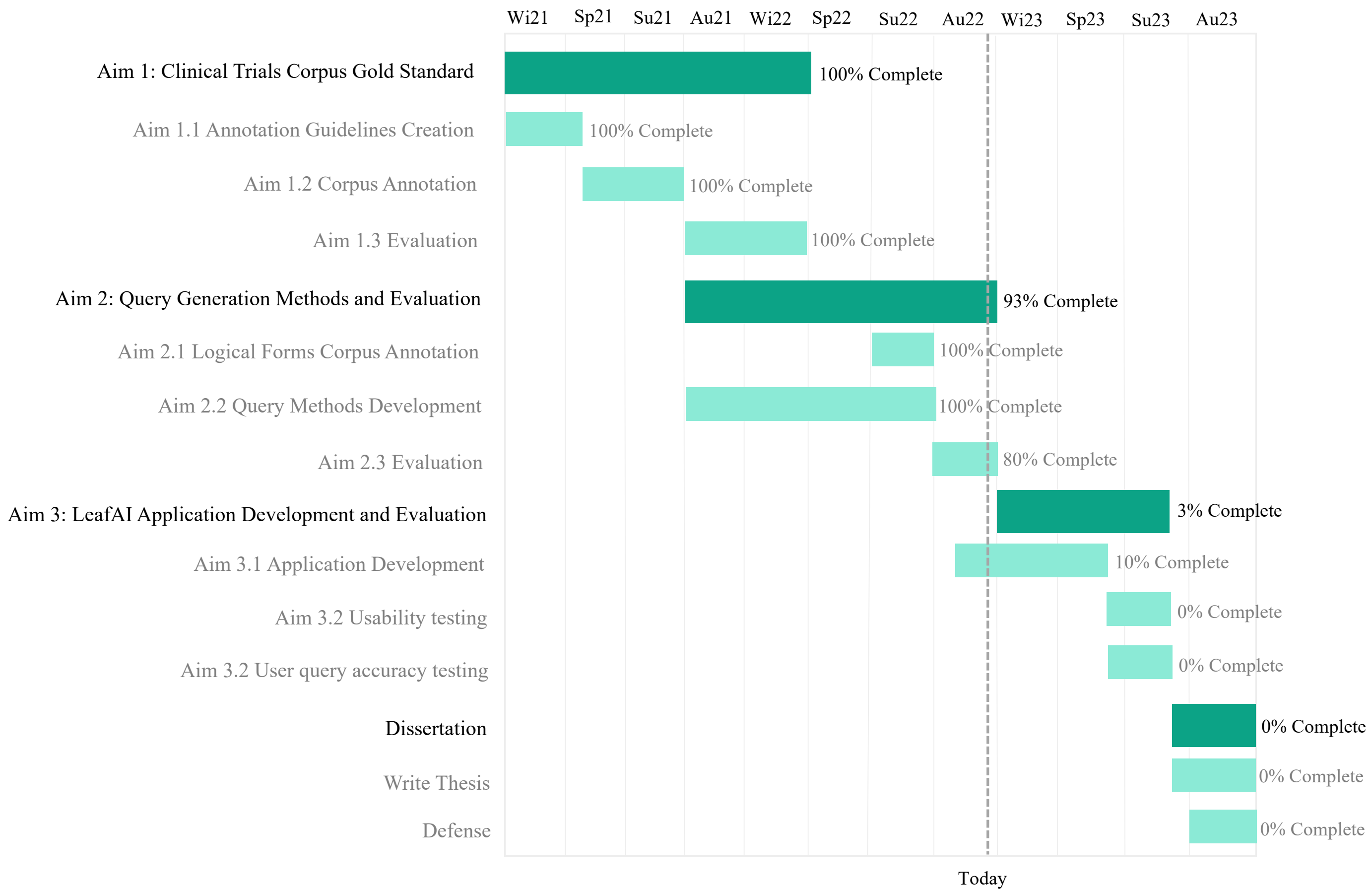


Figure 13: Project completion timeline.

Clinical trials have a significant impact on biomedical research and the approval of new drugs and methods for treating conditions which affect humans. Finding patients meeting criteria for clinical trials - and biomedical research in general - is often a time-consuming and difficult challenge. This project aims to use NLP and a user-friendly web application to help researchers accomplish this. In Aim 1, we developed the Leaf Clinical Trials corpus, a gold standard for identifying key elements within eligibility criteria. In Aim 2, we developed the Leaf Logical Forms corpus and state-of-the-art methods for generating queries to identify patients meeting eligibility criteria. In Aim 3, we will create an intuitive web application, LeafAI, capable of quickly executing user-defined criteria and transparently explaining its findings.

7 Acknowledgements

This study was supported in part by the National Library of Medicine under Award Number R15LM013209 and by the National Center for Advancing Translational Sciences of National Institutes of Health under Award Number UL1TR002319. Experiments were run on computational resources generously provided by the UW Department of Radiology.

References

[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*, 20(e2):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*, 27(1):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*, 17(2):124–130, 2010.

[4] 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*, 7(1):1–11, 2020.

[5] 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*, 24(6):1062–1071, 2017.

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

[7] P. M. Rothwell. External validity of randomised controlled trials:"to whom do the results of this trial apply?". *The Lancet*, 365(9453):82–93, 2005.

[8] G. Frank. Current challenges in clinical trial patient recruitment and enrollment. *SoCRA Source*, 2(February):30–38, 2004.

[9] J. D. Grill and J. Karlawish. Addressing the challenges to successful recruitment and retention in Alzheimer's disease clinical trials. *Alzheimer's research & therapy*, 2(6):1–11, 2010.

[10] 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*, 39(2):169–182, 2014.

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

[12] 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*, 6(6):610–617, 2009.

[13] L. Penberthy, R. Brown, F. Puma, and B. Dahman. Automated matching software for clinical trials eligibility: measuring efficiency and flexibility. *Contemporary clinical trials*, 31(3):207–217, 2010.

[14] 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*, 203(4):415–422, 2012.

[15] J. Sullivan. Subject recruitment and retention: barriers to success. 2004.

[16] 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*, 16(6):869–873, 2009.

[17] P. Easterbrook and D. Matthews. Fate of research studies. *Journal of the Royal Society of Medicine*, 85(2):71, 1992.

[18] 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*, 7(1):1–8, 2006.

[19] L. Marks and E. Power. Using technology to address recruitment issues in the clinical trial process. *Trends in biotechnology*, 20(3):105–109, 2002.

[20] 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*, 22(1):166–178, 2015.

[21] M. Sok, M. Zavrl, B. Greif, and M. Srpčič. Objective assessment of WHO/ECOG performance status. *Supportive Care in Cancer*, 27(10):3793–3798, 2019.

[22] 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*, 2(10):749–760, 2018.

[23] E. Jermutus, D. Kneale, J. Thomas, and S. Michie. Influences on User Trust in Healthcare Artificial Intelligence: A Systematic Review. *Wellcome Open Research*, 7:65, 2022.

[24] A. Sertkaya, H.-H. Wong, A. Jessup, and T. Beleche. Key cost drivers of pharmaceutical clinical trials in the United States. *Clinical Trials*, 13(2):117–126, 2016.

[25] 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*, 5, 2021.

[26] 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*, pages 86–94, 2021.

[27] 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*, pages 138–142, 2021.

[28] 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*, 96:103239, 2019.

[29] 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*, 36(2):119–134, 2013.

[30] 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*, 21(3):e13249, 2019.

[31] 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.

[32] P. Warrer, E. H. Hansen, L. Juhl-Jensen, and L. Aagaard. Using text-mining techniques in electronic patient records to identify ADRs from medicine use. *British journal of clinical pharmacology*, 73(5):674–684, 2012.

[33] 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*, pages 1029–1037, 2020.

[34] 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*, volume 2017, page 1754. American Medical Informatics Association, 2017.

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

[36] 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*, 18(Supplement 1):i116–i124, dec 2011.

[37] 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. (May):5829–5837, 2020.

[38] 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*, 216:574, 2015.

[39] 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*, pages 102–107, 2012.

[40] 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*, 2012:71–80, 2012. PMID: 22779055.

[41] A. X. Chang and C. D. Manning. Sutime: A library for recognizing and normalizing time expressions. In *Lrec*, volume 3735, page 3740, 2012.

[42] 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*, 26(4):294–305, 2019.

[43] 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.

[44] 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.

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

[46] 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)*, 3(1):1–23, 2021.

[47] 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*, pages 2361–2364, 2019.

[48] 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)*, pages 1532–1543, 2014.

[49] S. Soni and K. Roberts. Patient cohort retrieval using transformer language models. In *AMIA annual symposium proceedings*, volume 2020, page 1150. American Medical Informatics Association, 2020.

[50] 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.

[51] 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*, 26(11):1218–1226, 2019.

[52] 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*, pages 534–538. IOS Press, 2015.

[53] H. Dhayne, R. Kilany, R. Haque, and Y. Taher. EMR2vec: Bridging the gap between patient data and clinical trial. *Computers & Industrial Engineering*, 156:107236, 2021.

[54] 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*, 592(7855):629–633, 2021.

[55] 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*, 26(11):1203–1208, 2019.

[56] 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.

[57] 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*, pages 13–25. PMLR, 2021.

[58] 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*, pages 36–53. PMLR, 2021.

[59] 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*, pages 350–361, 2020.

[60] 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*, 9(12):e32698, 2021.

[61] 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*, 3(1):1–9, 2016.

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

[63] OWL 2 Web Ontology Language Primer. `https://www.w3.org/TR/owl2-primer/`. Accessed: 2022-08-16.

[64] 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*, pages 816–829. Springer, 2007.

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

[66] 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*, pages 11–25. Springer, 2013.

[67] 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*, pages 1069–1074, 2018.

[68] 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.

[69] 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*, volume 2016, page 3772. NIH Public Access, 2016.

[70] 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.*, 21(140):1–67, 2020.

[71] 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.

[72] C.-Y. Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81, 2004.

[73] 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*, pages 249–256, 2006.

[74] gRPC: A high performance, open source universal RPC framework. `https://grpc.io//`. Accessed: 2022-08-16.

[75] Docker. `https://www.docker.com`. Accessed: 2022-08-16.

[76] A. Johnstone and E. Scott. Generalised recursive descent parsing and follow-determinism. In *International Conference on Compiler Construction*, pages 16–30. Springer, 1998.

[77] 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*, 9(1):1–15, 2022.

[78] A. R. Aronson. Effective mapping of biomedical text to the UMLS Metathesaurus: the MetaMap program. In *Proceedings of the AMIA Symposium*, page 17. American Medical Informatics Association, 2001.

[79] 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*, 24(4):841–844, 2017.

[80] K. Lee and Ö. Uzuner. Normalizing Adverse Events using Recurrent Neural Networks with Attention. *AMIA Summits on Translational Science Proceedings*, 2020:345, 2020.

[81] 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*, 37(suppl_2):W170–W173, 2009.

[82] 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*, 40(D1):D940–D946, 2012.

[83] 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*, 39(suppl_1):D38–D51, 2010.

[84] 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*, 36(24):5703–5705, 2020.

[85] 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*, 55:206–217, 2015.

[86] 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*, 2(1):1–9, 2019.

[87] X. Wang, A. Chused, N. Elhadad, C. Friedman, and M. Markatou. Automated knowledge acquisition from clinical narrative reports. In *AMIA Annual Symposium Proceedings*, volume 2008, page 783. American Medical Informatics Association, 2008.

[88] 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*, pages 441–450, 1973.

[89] 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.*, pages 320–325. IEEE, 1978.

[90] 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*, volume 2, pages 255–261. IEEE, 1999.

[91] 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*, 44(2):277–288, 2011.

[92] M. Derthick and S. F. Roth. Enhancing data exploration with a branching history of user operations. *Knowledge-Based Systems*, 14(1-2):65–74, 2001.

[93] 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*, pages 431–438, 2004.

[94] W. C. Hill and J. D. Hollan. History-enriched digital objects: Prototypes and policy issues. *The Information Society*, 10(2):139–145, 1994.

[95] S. Li and C.-H. Chen. The effects of visual feedback designs on long wait time of mobile application user interface. *Interacting with Computers*, 31(1):1–12, 2019.

[96] 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*, pages 103–112, 2014.

[97] B. Shneiderman. Response time and display rate in human performance with computers. *ACM Computing Surveys (CSUR)*, 16(3):265–285, 1984.

[98] 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*, pages 19–28, 2003.

[99] 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*, pages 609–618, 2016.

[100] 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*, 30, 2017.

[101] 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*, 18(1):6869–6898, 2017.

[102] P.-Y. Yen, D. Wantland, and S. Bakken. Development of a customizable health IT usability evaluation scale. In *AMIA Annual Symposium Proceedings*, volume 2010, page 917. American Medical Informatics Association, 2010.