Toward a Natural Language Interface for EHR Questions

Kirk Roberts, PhD, and Dina Demner-Fushman, MD, PhD
Lister Hill National Center for Biomedical Communications
U.S. National Library of Medicine

Abstract

This paper presents a pilot study on the process of manually annotating natural language EHR questions with a formal meaning representation. This formal representation could then be used as a structured query as part of a natural language interface for electronic health records. This study analyzes the challenges of representing EHR questions as structured queries as well as the feasibility of creating a sufficiently large corpus of manually annotated structured queries for EHR questions. A set of 100 EHR questions, sampled from actual questions asked by ICU physicians[1], is used to perform the analysis. The ultimate goal of this research is to enable automatic methods for understanding EHR questions for use in a natural language EHR interface.

Introduction

Efficient access to medical information is important for patient care. Studies have found that between 27 and 53% of the questions doctors ask require patient-specific information to answer[2,3]. Information such as lab results, current medications, and past medical history are the basis for doctors to diagnose or treat patients, and for helping patients better manage their own health. The primary source for patient-specific medical information is the electronic health record (EHR). Efficient access to patient-specific data is thus a critical consideration in EHR design.

Natural language questions are an intuitive way to query EHRs. In observational studies, doctors typically expressed their information needs as questions[2,4]. These observations have spurred a significant amount of work in medical question answering (QA)[5], yet little work has been performed in question answering for EHRs. Currently EHRs are queried using keywords, and while this approach might provide access to needed information, it does not provide an exact answer and often requires navigating many retrieval results.

The primary challenge in applying a QA system to EHR data is understanding question meaning. This challenge is manifested in the process of converting a natural language question into an unambiguous structured query. In natural language processing (NLP), this process is referred to as *semantic parsing* and typically employs grammars that encode transformations between unstructured text and a structured representation. For instance, the grammar might encode that "*how many X*" corresponds to a function that counts the number of items in X. Manually constructing such grammars is extremely difficult, so they are typically learned automatically from corpora consisting of $<$*unstructured question*, *structured query*$>$ pairs. Even this manual annotation process, however, is a difficult, time-consuming task. Existing semantic parsing corpora are typically quite small (a few hundred questions) and are limited to a very specific domain (e.g., geography, airline flight information). Constructing a corpus of sufficient size and quality for understanding EHR questions presents a significant undertaking, so in this work we conduct a pilot study to determine what issues need to be addressed in the structured representation and manual annotation of EHR questions. The specific contributions of this work are:

(1) an assessment of the feasibility of creating a large corpus for semantic parsing, including an analysis of how many EHR questions are not amenable to a structured form,

(2) a discussion of the linguistic challenges in representing EHR questions,

(3) a proposed structured representation that is independent of any particular EHR design, and instead based on a form of first order logic commonly used with semantic parsing methods, and

(4) a set of 100 EHR questions manually annotated with the proposed logical forms.

Note that such a structured query is not only independent of any particular EHR, but also the way in which an EHR stores its data. While an unstructured query (keywords) cannot be used for searching structured data, a structured query can easily be converted to operate over unstructured clinical narratives.

	admission	discharge	PMH	visit	status	plan	time range	problem	treatment	test
Li2012	3.1	1.1	1.3	95.1	31.4	1.1	11.6	19.2	35.2	35.6
100 Q	5.0	2.0	3.0	92.0	25.0	2.0	13.0	23.0	34.0	31.0

	boolean	count	trend	medical	time	person/org	other	list
Li2012	47.9	9.1	0.9	31.9	8.9	0.7	0.7	3.1
100 Q	42.0	11.0	2.0	31.0	12.0	2.0	0.0	7.0

Table 1: % of questions for each category in the Li question set and the sample for this study. A question can be in more than one category.

Background

Natural language questions are often used to express clinical needs. Typically, studies of these questions have not focused on how they can be answered exclusively by the EHR, but rather whether the information resources exist at all in the clinical setting to provide answers. Our work focuses entirely on patient-specific questions that are answerable by the EHR. A related approach for EHR question understanding is Patrick and Li[6], who propose a method to map questions to a template. For example, the question "*Do they have an infected foot?*" is assigned the template [DID PATIENT (HAVE) X, Y?], while the question "*Is his abdomen distended?*" is assigned [DOES THE PATIENT'S X HAVE (BEEN) Z?]. There are several difficulties with this template-based approach. First, individual classifiers are necessary to extract the relevant X, Y, and Z values (and in many cases no such value exists). Second, if machine learning (ML) is used for classification, the templates result in unnecessary data segmentation, so many questions must be annotated for each template. Third, the templates impose unnecessary constraints, as slight changes to a question might require a new template be created, as well as additional classifiers and data. As a result, the template approach scales exponentially with the number of possible constraints, while grammar-based approaches only scale linearly in this respect. Finally, grammar-based approaches typically *know what they do not know* (i.e., they recognize a question is beyond the scope of a grammar), while a classification approach will typically chose the closest (wrong) class, resulting in a degraded user experience. These difficulties suggest that the top-down process of manually defining templates cannot scale to the range of possible EHR questions. Instead, we propose a bottom-up grammar-based technique that builds a structured query from individual question phrases.

Beyond the focus on EHR questions, some research has created logical forms from medical questions[7,8], but they employ "shallow" logical forms based on the words in the syntactic dependency tree. Instead, "deep" semantic parsing requires transforming questions using an existing knowledge base (KB) of logical operations, though this KB can be created directly from the training data. This well-defined set of operations is important to create a structured query that can be interpreted by third parties. Deep semantic parsing of questions has seen significantly more attention in other domains, such as baseball statistics and geography[1]. The semantic parsing approaches to these datasets differ in their logical form, grammatical structure, and learning constraints. In this pilot study, we focus on only on the issue of representation, so the precise grammar structure and learning method need not be defined. Instead, our goal is to test the feasibility of logical forms to represent EHR questions, largely along the lines of lambda calculus[9]. If these logical forms cannot represent the user's request, then the other considerations would be irrelevant.

Methods

1 Question Set

To obtain EHR questions for analysis, 100 questions from the Li[1] question set (referred to as Li2012) were sampled as described below. The Li2012 questions were collected from physicians in an Intensive Care Unit in an Australian hospital. These questions are organized into several groups, only two of which were identified as containing patient-specific EHR questions: structured database (17 questions) and specific note (432). The other groups include external knowledge questions or high-level reasoning questions that would not be directly answerable by the EHR.

In order to ensure that the 100 sampled questions were representative of the phenomena found in patient-specific EHR questions, a categorization scheme was created and manually annotated by both authors. This also serves as a useful analysis for the types of questions ICU physicians might ask a natural language EHR interface. Questions were categorized based on temporal information (admission, discharge, past medical history, current visit, current status, plan, and explicit time range), clinical information (problem, treatment, and test), answer type (boolean yes/no, count, trend, medical code/measurement, time/date, person/organization, or other), and if the answer is intended to be a list or single response. The goal of the sampling was to produce 100 questions representative both of these categories and

the templates assigned in Li2012. The Li2012 questions are already sorted by template and then alphabetically, so to get a representative sample of the template types, every tenth question was included in our sample. Then, the rarer categories were over-sampled to create a data set that is more balanced than the original 449 questions. Table 1 shows the percentage of questions in each category for both the full 449 questions and the sampled 100 questions.

2 Preprocessing

To reduce the complexity of semantic parsing and leverage other NLP components, a preprocessing stage is used to normalize common EHR question concepts. First, references to the patient (e.g., *he*, *she*, *the patient*) are replaced with `patient`. Similarly, admission and discharge references are normalized. Next, temporal and spatial expressions are replaced with placeholders. Then, UMLS concepts are normalized to their CUI and semantic type. The output of this step is a question that containing the same information as the original, but generalized for easier parsing. For example:

> Was she hypertensive on admission?
> Was `patient finding`("Hypertensive", `C0857121`) on `admission`?
>
> Did he have any diabetes?
> Did `patient` have any `disease`("Diabetes Mellitus", `C0011849`)

Each of the 100 sample questions was manually processed this way. The best UMLS concept CUI to choose for a given term was not always straight-forward. The *Discussion* section addresses this linguistic challenge in further detail.

3 Structured Representation

To represent questions as a structured query, a variant of first order logic (FOL) is used that combines FOL with lambda calculus expressions. This type of logical form is commonly used for semantic parsing[9] since it is a very intuitive way to interpret questions while still having a fairly compact form. FOL combines atomic objects (e.g., `Richard`, `John`), boolean-valued predicates (e.g., *is_king*(`Richard`)), and functions (e.g., *mother*(`Richard`) returns `Eleanor`). When incorporating lambda calculus, the traditional FOL quantifiers $\forall$ and $\exists$ are generally replaced with the λ quantifier that denotes a set. For instance, $\lambda x.$*is_king*(x) is the set of all objects for which *is_king* is true. When applied to EHRs, predicates operate on medical events. For instance, given the objects `C0004057` (the UMLS CUI for aspirin) and `visit` (which corresponds to all EHR information about a patient's current hospital visit) a simple logical form can express all events of the patient taking aspirin: $\lambda x.$*has_treatment*(x, `C0004057`, `visit`). The formal interpretation of a question is then the logical form that describes the question's answer.

4 Logical Form Annotation

To decide on an initial set of objects, predicates, and functions, both authors annotated 25 of the questions with logical forms and then came to an agreement on their basic structure. The core of the logical forms are typically structured as $\{\lambda x.$*has*_TYPE(x, CODE, TIME)$\}$, where TYPE is a medical class (e.g., treatment, disease, device); CODE is the UMLS CUI for the medical concept; and TIME is the temporal boundary (past medical history, current visit, current status, future plan). These sets can then be modified with functions like *count* (the total number of items in a set) and *latest* (the item in a set with the most recent timestamp), or other predicates such as δ (which is true if a set is non-empty). For instance, $\{$*latest*($\lambda x.$*has_test*(x, `C0392201`, `history`))$\}$ is the most recent blood glucose measurement, while $\{\delta(\lambda x.$*has_treatment*(x, `C0005841`, `visit`))$\}$ indicates whether or not a blood transfusion was performed during the visit. The authors then double-annotated and resolved disagreements on the next 25 questions, achieving a 40% agreement on complete logical forms and a 81% agreement on individual logical form elements. This suggests that while there was low exact agreement, the annotators agreed on most parts of the logical form. Finally, the final 50 questions were double-annotated and reconciled. This set achieved a 50% agreement on complete logical forms and 85% agreement on individual logical form elements, demonstrating an increasing agreement on this task.

Results

Based on our annotation process, 113 logical objects, 136 predicates, and 226 functions were used to represent the 100 questions. The frequency distribution of the elements suggests there is a very long tail of potential logical elements (e.g., the predicate *is_positive* was used only once, while its inverse *is_negative* was never used but certainly could be). This suggests that a complete set of logical elements would be difficult to find, as new questions might introduce new functionality not addressed by the existing standard. However, the vast majority of questions can be logically encoded with a very small set of elements (86% of the annotated elements are composed of only 12 unique elements).

All 100 questions were represented in a logical form, though sometimes the logical form might have a different interpretation than what was originally intended. The larger issue of literal versus intentional meaning is addressed in the *Discussion*, but a specific example of a question that might be difficult to convert to a logical form is as follows:

LEXICAL: How does his chest sound?
CONCEPT: How does patient's vb:finding('Sounds within the chest', C0425538)?
LOGICAL: *latest*(λx.*has_finding*(x, C0425538, status))

Here, the logical form specifies the most recent finding of type C0425538 ("Sounds within the chest") which is not necessarily a direct description of the sound, though could contain it if such information is within the EHR. The difficulty of this question is that does not ask for a specific property but rather a domain-dependent description. However, assuming that a reference to the most recent finding (which may be associated with such a description) is a sufficient logical representation, then all 100 questions were representable with logical forms.

Discussion

Both the manual and automatic annotation of logical forms requires overcoming several linguistic challenges. First, deciding which UMLS code to use for a particular medical concept is not always straight-forward. For example:

(1) Did he awaken?
δ(λx.*has_finding*(x, C0234422, visit))
(2) What feeds is the patient on?
λx.*has_treatment*(x, C0596440, status)

In Question (1), we chose UMLS concept C0234422 ("Awake"), but also could have chosen C1821422 ("Fully awake") or C2051412 ("patient appears awake"). In Question (2), we chose C0596440 ("diet route/schedule"), which is too general, but the next best concept was C0086225 ("Enteral Feeding"), which is too specific. This problem stems from a combination of (a) a lack of specificity in the question, and (b) potential differences in the codes different EHRs use to encode the same medical idea. To some extent, the first problem can be solved by forcing users to be more specific, while the second might require knowing the valid set of codes prior to performing concept normalization. A related problem is metonymy, when one concept is used in place of another functionally-related concept:

(3) Has he had blood products?
δ(λx.*has_treatment*(x, C0852255, visit))

In Question (3), the physical substance (*blood products*) is used to refer to the treatment procedure that involves the substance. A different UMLS concept expresses the substance (C0456388) than the treatment procedure (C0852255). Recognizing the metonymy is important, as without properly interpreting it the question would be asking whether the patient physically possessed the substance instead of being treated with it.

A good amount of consideration went into how best to handle temporal and spatial references. Consider:

(4) What was the volume of his urine last night?
δ(λx.*has_function*(x, C0232856, visit) $\wedge$ *time_within*(x, "last night"))
(5) When was the patient first discharged from the ward?
time(*earliest*(λx.*has_event*(x, discharge, visit) $\wedge$ *at_location*(x, "the ward")))

Both questions contain relative references: to understand Question (4) one needs to know the current date/time, while to understand Question (5) one needs to know the current ward. Encoding phrases such as "*last night*" in FOL can become problematic, while location references are dependent on a specific environment. In both cases we decided to recognize the boundaries of the relative references and leave them for downstream handling. For instance, it is likely easier to convert "*last night*" to a range of absolute values, but that again requires knowing the current date/time.

One of the most difficult NLP tasks, within the field of pragmatics, is determining an author's intentions. This becomes an issue with EHR questions when the literal question asks for something different than what the user expects:

(6) Did she have a reaction to the treatment?
δ(λx.*has_treatment*(x, C0087111, visit) $\wedge$ (λy.*is_reaction*(x, y)))
(7) Has the blood pressure dropped?
is_decreasing(λx.*has_finding*(x, C1271104, visit)))

Both the above are yes/no questions, but only on the surface. In Question (6), the user probably wants to know what

reactions occurred, while Question (7) is looking for a more qualitative description of recent blood pressure measurements as opposed to whether a very minor drop occurred. In these cases, we chose to annotate the literal interpretation, both because automatically interpreting intentions would be difficult and because the user could rephrase the question to match the intention (e.g., "*What reactions did she have to the treatment?*").

Finally, natural language often assumes a high degree of domain knowledge. Consider:

(8) What are the positive tests?
positive(λx.*has_test*(x, `C0022885`, `visit`))

Here, *positive* filters out tests that are negative, but that depends on the type of test. Tests have different thresholds and standards that determine whether they are positive or negative. While this type of domain-specific function is perfectly fine for first order logic, it still means the domain information must be encoded in the (structured) query processing. Alternatively, users can be forced to specify their own values/thresholds (e.g., "*Is his A1c over 6.5?*").

Beyond the issue of representing a natural language question as a logical form, the difficulty of automatically converting a question to that form must also be considered. As can be seen in the examples above, the core medical predicates (*has_treatment*, *has_problem*, etc.) are typically quite predictable from the concepts in the question. The additional logical elements are largely based on the presence or absence of a specific phrase (e.g., *Did* becomes δ) or verb tense (e.g., present tense often indicates `status`). Without a corpus of sufficient size to train a semantic parser it is difficult to say how accurate a model can be built, but assuming many of the underlying linguistic issues can be accurately addressed, the potential for high-performing semantic parsing appears quite promising.

Conclusion

This paper presented a pilot study on manually annotating natural language EHR questions with a formal meaning representation based on first order logic with lambda calculus. The study analyzed the challenges of representing EHR questions as structured queries and the feasibility of creating a sufficiently large corpus of manually annotated EHR questions for the development of an automatic semantic parser. The results of the study demonstrated the feasibility of creating such a resource, potentially enabling the development of a natural language EHR interface.

Acknowledgements This research was supported by the Intramural Research Program of the National Library of Medicine, NIH.

References

1. M Li. *Investigation, Design and Implementation of a Clinical Question Answering System*. PhD thesis, University of Sydney, 2012.
2. JW Ely, JA Osheroff, MH Ebell, ML Chambliss, DC Vinson, JJ Stevermer, and EA Pifer. Obstacles to answering doctors' questions about patient care with evidence: qualitative study. *BMJ*, 324:1–7, 2002.
3. LM Currie, M Graham, M Allen, S Bakken, V Patel, and JJ Cimino. Clinical Information Needs in Context: An Observational Study of Clinicians While Using a Clinical Information System. In *AMIA Annu Symp Proc*, 2003.
4. G Del Fiol, TE Workman, and PN Gorman. Clinical Questions Raised by Clinicians at the Point of Care: A Systematic Review. *JAMA Intern Med*, 174(5):710–718, 2014.
5. SJ Athenikos and H Han. Biomedical question answering: A survey. *Comput Meth Prog Bio*, 99:1–24, 2010.
6. J Patrick and M Li. An ontology for clinical questions about the contents of patient notes. *J Biomed Inform*, 45:292–306, 2012.
7. RM Terol, P Martinez-Barco, and M Palomar. A knowledge based method for the medical question answering problem. *Comput Biol Med*, 37(10):1511–1521, 2007.
8. SJ Athenikos, H Han, and AD Brooks. A framework of a logic-based question-answering system for the medical domain (LOQAS-Med). In *ACM Symposium on Applied Computing*, pages 847–851, 2009.
9. L Zettlemoyer and M Collins. Learning to Map Sentences to Logical Forms: Structured Classification with Probabilistic Categorical Grammars. In *Conference on Uncertainty in Artificial Intelligence*, 2005.