Accessing Structured Health Information through English Queries and Automatic Deduction

Richard Waldinger[1], Daniel G. Bobrow[2], Cleo Condoravdi[2], Amar Das[3], Kyle Richardson[2],

[1] SRI International, [2] PARC, [3] Stanford University
waldinger@ai.sri.com, {bobrow, condorav, krichard}@parc.com, das@stanford.edu

Abstract

While much health data is available online, patients who are not technically astute may be unable to access it because they may not know the relevant resources, they may be reluctant to confront an unfamiliar interface, and they may not know how to compose an answer from information provided by multiple heterogeneous resources. We describe ongoing research in using natural English text queries and automated deduction to obtain answers based on multiple structured data sources in a specific subject domain.

Each English query is transformed using natural language technology into an unambiguous logical form; this is submitted to a theorem prover that operates over an axiomatic theory of the subject domain. Symbols in the theory are linked to relations in external databases known to the system. An answer is obtained from the proof, along with an English language explanation of how the answer was obtained. Answers need not be present explicitly in any of the databases, but rather may be deduced or computed from the information they provide.

Although English is highly ambiguous, the natural language technology is informed by subject domain knowledge, so that readings of the query that are syntactically plausible but semantically impossible are discarded. When a question is still ambiguous, the system can interrogate the patient to determine what meaning was intended. Additional queries can clarify earlier ones or ask questions referring to previously computed answers.

We describe a prototype system, Quadri, which answers questions about HIV treatment using the Stanford HIV Drug Resistance Database and other resources. Natural language processing is provided by PARC's Bridge, and the deductive mechanism is SRI's SNARK theorem prover. We discuss some of the problems that must be faced to make this approach work, and some of our solutions.

Introduction

There is a significant body of health information available in the form of structured databases. People who want to get access to this information may not know of the existence of the appropriate databases, or how to combine the accessed data to construct answers to questions. Using natural English as a user interface language would provide a patient with a way of interacting with information stores in a familiar way. Among the obstacles are that the machine understanding of natural language is a known hard problem and that natural languages are highly ambiguous. The thesis of the present work is that understanding a subject domain will mitigate the difficulties in handling English queries in that domain.

Answers are composed or inferred from multiple online structured knowledge sources, which can be heterogeneous—they may have been developed by different people or organizations, they may not have been intended to be used together, and they may have adopted no common representation. Such sources include databases, clinical records, formal ontologies, and software services. The vocabulary in which the query is expressed may be completely different from that employed by the relevant sources, and the patient may have no idea how each source is organized. Patients will be able to further specify a query, alter a previous query, or ask follow-up questions.

While the approach we adopt is domain independent, it requires building on a particular well-understood subject domain, so that domain knowledge can be brought to bear in understanding the question. Our prototype, Quadri, is directed towards the domain of HIV treatment, and especially drug resistance and changes of drug regimen. Quadri is intended to be used by researchers and physicians, but the technology is appropriate for use by patients, who may not have the time or expertise to become familiar with multiple data resources. Of course, making the data accessible to patients presents new obstacles.

Previous Work

The use of English as an interface language has a long history. One of the earliest efforts was the Lunar system [Woods, 1972], which uses a semantic grammar to analyze queries over a database about Lunar rocks. The START system [Katz, 2002] answers a query from multiple sources by matching the syntactic parse of the query with parses

associated with the sources. Quark [Waldinger 2004], the system closest to our own, uses the Gemini parser to create a logical form of the input query and the SNARK theorem prover to create answers to this query. SRA from Cycorp [Lenat 2010] also provides an English front end to a reasoning system that can access structured data; primarily using local domain constraints, it incrementally and interactively refines the interpretation of a user query.

Our projected work is distinguished from these by

- using language analysis that does not prematurely eliminate syntactic ambiguity, but that rather preserves it in a compact form, similarly to the Core Language Engine [Alshawi 1992].
- using domain knowledge to prune the ambiguities, during both the language analysis and the search for answers.
- generating a logical form for a query that captures the logical dependencies and that uses a higher level vocabulary that can be interpreted by axioms of the domain theory.
- enabling users to extend, refine, and alter their questions using a *stream* of queries, and to ask follow-up questions that use the results of preceding queries.
- giving feedback on a query's logical interpretation and an explanation of how the answer was obtained.

English to Logical Form

Our approach combines natural language with deductive technology. Each query is transformed into a logical form, which is submitted to a theorem prover provided with an axiomatic theory of the subject domain. The parser is equipped with subject domain vocabulary and some of the sort (type) hierarchy of the subject domain. Often questions that are ambiguous syntactically are unambiguous when the subject domain is understood. For example, in the sentence: "Find a patient on Atripla with the mutation M184V." the phrase "*with the mutation M184V*" can syntactically modify *Atripla* or *patient*. However, Atripla is a drug combination, and M184V is a mutation. Because drugs don't mutate, M184V must be syntactically linked to the patient, not the drug. Furthermore, domain knowledge, expressed in the axiomatic theory, states that when asked for a patient with a particular mutation, look for a mutation present in the gene of the HIV virus that was obtained from the patient, not in the patient directly, even though no virus is mentioned.

Quadri is sensitive to the logical structure of a query and knowledgeable about temporal relations—we might want to find patients who exhibited a high viral load near the end of a twenty-four week regimen, or a patient who has been on Atripla for at least eight weeks.

To provide user feedback, the constructed logical form is rephrased as a pedantic English sentence that is close to the logic. If there is more than one possible interpretation, the user may be asked to choose among alternative phrases, or to rephrase the question in a less ambiguous way. This provides assurance that the user's query was correctly understood.

Inference

The logical form produced by the natural language component is presented to a theorem prover, which tries to prove it in the axiomatic subject-domain theory. This theory contains axioms that define the meaning of the concepts in the query, express the capabilities of the various knowledge sources, and provide the background knowledge necessary to link them together. The query theorem is transformed and decomposed according to the axioms of the theory. When a deduced subquery is simple enough to be answered directly by a single knowledge source, such as a database or a software service, that source is queried as the proof is underway, via a mechanism known as *procedural attachment*. In this way, information provided by the source can be used in the proof, even if it is not mentioned in any axiom.

When the proof is complete, an answer to the original query can be constructed, via an *answer-extraction* mechanism. The explanation of how the answer was obtained and the provenance of its data are also extracted from the proof.

Typically the query requires us to find entities that satisfy specified conditions. The theorem prover will prove the existence of such entities by finding patient records with the correct properties; information from these records can then be introduced into the proof by the procedural-attachment mechanism.

Because of the heterogeneity of the knowledge sources, it can happen that the form of the data produced by one source differs from the form required by another. In that case, a translation software service is invoked, by the same procedural-attachment mechanism.

Answers need not be collections of records; they may be complex structures that contain tables and visualizations. We are coordinating with Stanford's SweetInfo project, which constructs visualizations of HIV Data.

We are exploring these ideas by developing the prototype system Quadri for the HIV application. Quadri consists of a natural language component, a deductive component, and the relevant data sources. We examine these in the next two sections.

Bridge

Quadri is based on Bridge [Bobrow 2007], a general natural-language processing system. Bridge consists of a number of language-processing modules that can be customized for a specific domain. Text is analyzed with a finite-state machine that recognizes named entities and standard English morphology, and can be augmented to recognize specialized notation, such as M184V as a mutation. The syntactic parser, XLE [Maxwell 1996], uses a broad-based English grammar tunable through training. It produces dependency analyses of a sentence, using a compact notation to capture ambiguities. Rewrite rules take

this nested dependency structure and produce a flattened semantic representation [Crouch 2005] in which alternative expressions (e.g., passive and active sentences) are mapped to a common representation in a knowledge-representation language [Bobrow 2007].

Relations among terms are captured in an ontology, with domain-specific synonyms and sort structure augmenting a broad-based English lexicon (WordNet). The nesting of quantifiers for the logical form is extracted from this representation. Mapping the English structures into domain-specific relations, the Quadri system eliminates syntactic ambiguities that are not interpretable in those terms. For example, "the patient had a high viral load" is mapped into the relation:

patient-has-test(patient, viral-load, high, time),

where **time** is the time at which the test occurred.

For later sentences in the query stream, definite and anaphoric references are marked for the theorem prover to fill in the values. Information is available about the sort and the positions of possible targets for the reference. Filling in these references is not yet complete.

SNARK

The deductive component of Quadri consists of the theorem-proving system SNARK with an axiomatic theory of HIV drug resistance. Procedural attachments access the Stanford HIV Drug Resistance Database [Rhee et al 2003, Shafer 2006].

SNARK [Stickel et al. 2000] is a first-order-logic theorem prover developed at SRI. It contains many of the most successful inference mechanisms for general-purpose automated reasoning (sorted resolution, paramodulation, rewriting, etc.) plus procedures that perform accelerated special-purpose inference (e.g., numerical computation and temporal and spatial reasoning). SNARK has devices for procedural attachment and answer extraction. It has strategic control features that allow us to tailor it to exhibit high performance in a selected subject domain. It is mature software that has been applied to a number of successful applications (e.g., NASA's Amphion [Stickel et al. 1994], which analyzes data from space missions, uses SNARK.)

An Example

Let us consider an example. Suppose Quadri is given the query *Find patients who had a high viral load after almost 24 weeks on a regimen with EFV and 3TC*.

Note that merely understanding this query requires considerable subject domain knowledge. We must recognize that *EFV* (Efavirenz) and 3TC (Lamivudine) are drugs, that *viral load* is a medical test, that *high* is the result of the test, that the drugs are with the regimen (and not with the viral load test, say, or the patient) and that *after 24 weeks* refers to the approximate time of the medical test since the beginning of the regimen. Given the appropriate knowledge (regimens have drugs and temporal extent; drugs do not have a temporal dimension, etc), Quadri is able to generate a logical form.

This formula is a conjunction of several conditions, such as

patient-has-regimen(?patient, ?regimen), and
regimen-has-drug-set(?regimen, set(efv, 3tc)).

In other words, the patients must satisfy the condition that they each have a regimen that contains the specified drugs. The symbols with question marks have tacit sorted existential quantification; that is, we must find patients and regimens that satisfy these conditions during the proof search. Once we have proved that such patients exist, the answer-extraction mechanism will be able to produce the exemplars that have been found during the proof process.

Other conditions in the logical form are the following:

patient-has-test (?patient, viral-load, high, ?time2),
starts-time(?time1, ?regimen),
starts-time(?time1, ?interval),
finishes-time(?time2, ?interval), and
almost(duration(?interval), weeks(24)),

In other words, the patient must have a high viral load almost at the end of the twenty-four-week interval that starts at the beginning of the regimen. The temporal relation

starts-time(?time, ?interval)

holds if **?time** is the initial time-point of **?interval**.

The formula is submitted to SNARK to be proved as a theorem. Axioms of the subject domain theory allow SNARK to relate the abstract, approximate, qualitative relations in the query to concrete, exact, quantitative ones that have a direct representation in the database. For instance, the axiom

patient-has-test(?patient, ?test, ?result, ?time2)
⇐
hiv-db-test(?patient, ?test, ?measurement, ?time)
& qual-viral-load(?measurement, ?result)
& near(?time, ?time2)

states that, for the test to yield a qualitative result (e.g, *high*) at an approximate time **?time2,** it must yield a precise numerical measurment (e.g., 5) at a precise time **?time** that is "near" **?time2.** Other axioms tell us that the relation **qual-viral-load** holds for result *high* if the measurement is within a specified range. Two quantities are defined to be **near** each other if they are half a unit apart, where (in this theory) the unit of time is taken to be the week. A quantity is **almost** another if it less than the other, but more than 90 percent of it.

Some of the relations (e.g., **patient-has-test**) allow the use of qualitative values; others (e.g., **hiv-db-test**) refer to the quantitative values stored in the database, and are equipped with procedural attachments that can consult the database on the fly, as the proof is under way. Thus, if we are considering particular patients, procedural attachments will yield their regimens and tests; for each regimen, another procedural attachment will yield the drugs in that regimen, and others will yield its start and finish dates.

During the proof, procedural attachments reveal that a patient Mr. A2 (not his real name) has a second regimen

that begins on January 1, 2008. That regimen consists of the drugs 3TC, AZT (Zidovudine), and EFV. On June 17 of that year, he had a test that indicated a viral load of 5 (which is high). Because the drug regimen contains EFV and 3TC, and because the time of the test is almost twenty-four weeks after the start of Mr. A2's second regimen, SNARK is able to prove the theorem; the answer-extraction mechanism yields Mr. A2 as one of the requested patients, and the explanation mechanism is able to produce sentences similar to the ones we just used in paraphrasing the proof to justify the answer; these sentences are constructed from the axioms used in the proof. Other proofs yield other exemplars.

Query Streams and Anaphoric Reference

The above example dealt with a single question. It is more useful if Quadri can deal with a stream of queries, each of which may elaborate on the previous ones and refer back to their results. This extension is our next step.

For instance, the above question might have been phrased as a stream of queries; at first, the user requests *Find patients on a regimen containing EFV and 3TC*. Seeing the data, the user might then ask *Which of those had a high viral load after almost twenty-four weeks?*

Note that Quadri must understand that *those* in the second query refers to the patients, not to regimens or drugs, because patients can have high viral loads, but regimens and drugs cannot. If we follow with a query *Which of them also had AZT,* the word *them* must be taken to refer to the regimens of the patients, not the patients themselves, because, in our subject domain theory, regimens, not patients, have drugs.

Sometimes the resolution of such references is more subtle. For instance, if the last query was *For which of them did the failing regimen contain AZT?* Quadri must understand that *the failing regimen* refers to the regimen in which the high viral load was detected. On the other hand, if the new query was *For which of them did the salvage regimen contain LPV (Loprinavir)?* Quadri must understand that the salvage regimen is not the failing regimen, but the one that replaces it.

Status

We have collected a corpus of queries from our partners at the Stanford Biomedical Informatics group. The Quadri prototype is now capable of handling queries at the level of our principal example. It provides feedback to the user of the translation of the logical form, can prove the associated theorems, and can query a snapshot of the database to identify cohorts of patients that satisfy stated user criteria. Next steps include handling anaphoric references, enabling users to provide feedback to choose among alternative interpretations, and dealing with sequences of questions. Even in its present form, Quadri has impressed us and our collaborators with its ability to handle complex ambiguous constructions.

Acknowledgements

Robert Shafer and Soo-Yon Rhee provided their expertise on HIV and the Stanford HIV Drug Resistance Database. Mark Stickel assisted us with the use of the SNARK theorem prover. Will Bridewell and Cindy Mason provided comments and suggestions. The project described was supported by Award Number RC1LM010583 from the National Library of Medicine. The content is the sole responsibility of the authors and does not necessarily represent the official views of the National Library of Medicine or the National Institutes of Health.

References

Alshawi, H. (ed.) 1992. *The Core Language Engine*. MIT Press.

Bobrow, D. G. et al. 2007. PARC's Bridge and Question Answering System, *Proceedings of the Grammar Engineering Across Frameworks (GEAF07) Workshop*, pp. 46-66, CSLI Publications.

Crouch, R. 2005. Packed Rewriting for Mapping Semantics to KR. *Proceedings of the Sixth International Workshop on Computational Semantics,* Tilburg, The Netherlands.

Katz, B. et al. *2002*. The Start Multimedia Information System: Current Technology and Future Directions. *Proceedings of the International Workshop on Multimedia Information Systems*.

Lenat, D. et al. 2010. Harnessing Cyc to Answer Clinical Researchers' Ad Hoc Queries, *AAAI Magazine*.

Maxwell, J. T. and Kaplan, R. 1996. An efficient parser for LFG. Butt, M. and King, T. H. (eds.), *On-line Proceedings of the LF G96 Conference*. http://csli-publications.stanford.edu/publications.

Rhee, S. Y. et al 2003. Human immunodeficiency virus reverse transcriptase and protease sequence database. *Nucleic Acids Res* 31:298-303.

Shafer, R. W. 2006. Rationale and Uses of a Public HIV Drug-Resistance Database, *Journal of Infectious Diseases* 194 Suppl 1:S51-8.

Stickel, M et al. 2000. *A Guide to SNARK*. www.ai.sri.com/snark/tutorial.html

Stickel, M. et al. 1994. Deductive composition of astronomical software from subroutine libraries. *Proceedings of the Twelfth International Conference on Automated Deduction,* Nancy, France.

Waldinger, R. et al. 2004. Deductive Question Answering from Multiple Resources. *New Directions in Question Answering,* AAAI Press

Woods, W. A. et al. 1972. The Lunar Sciences Natural Language Information System: *Final Report, BBN Report No. 2378. Cambridge, MA 02138*. (Available from NTIS as N72-28984).