Answering Questions about HIV Drug Resistance using Natural Language Technology and Theorem Proving

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

[1] Palo Alto Research Center
3333 Coyote Hill Road, Palo Alto, CA, USA
{condorav, krichard, bobrow}@parc.com
[2] Stanford Center for Biomedical Informatics Research
251 Campus Drive, X215, Stanford, CA, USA
das@stanford.edu
[3] SRI International
333 Ravenswood Avenue, Menlo Park, CA, USA
waldinger@ai.sri.com

Abstract. We present preliminary work on an intelligent interface for answering English language clinical queries. Although our approach is domain independent, we focus on the needs of clinical researchers who are identifying cohorts of patients based on HIV drug-resistance patterns. Such questions are transformed into an unambiguous logical form by natural language technology (Bridge), which is then sent to a theorem prover (SNARK) that operates over an axiomatic theory of the subject domain. Symbols in the theory are linked to relations in one or more knowledge resources, such as databases, and an answer is obtained from the proof. Answers may be deduced or computed if they are not represented explicitly in a resource. We describe the status of our prototype system, called Quadri. We discuss some of the challenges that need to be solved to make this approach work and present some of our solutions.

Keywords: Temporal querying, natural language technology, deductive reasoning, axiomatic domain theory, drug resistance.

1 Introduction

Clinical researchers frequently need to access information in knowledge and data sources, but may lack familiarity with how information in a source is structured. To make such question answering easier, we are working on a prototype system, called Quadri, which uses a natural language English interface to request clinical information.

1.1 HIV Drug-Resistance Domain

Although our approach is domain independent, our initial application has been to assist investigators studying HIV drug resistance. Despite the success of highly active retroviral combination therapy in reducing morbidity and mortality associated with HIV, drug resistance remains a major obstacle to successful treatment. Clinical researchers use time-stamped data on drug regimens, HIV genotype tests, and viral loads to determine which HIV mutations emerge when a drug is administered and whether the presence of a new mutation is associated with poor drug response.

Investigating such questions requires evaluating temporal relations among the data, such as whether a mutation appeared after a particular drug was given or whether a high viral load occurred two months after the start of a regimen. Researchers may also inquire about more complex temporal patterns that link a baseline genotype test result measured at the end of a failing regimen with a viral response in its follow-up regimen. Consider the following example query:

> *Find patients who had a high viral load after almost 24 weeks on a regimen with EFV and 3TC.*

Understanding the content of this query requires considerable knowledge of the subject domain. We must recognize that *EFV* (Efavirenz) and *3TC* (Lamivudine) are three-letter abbreviations of drug names, that *viral load* is a laboratory test, that *high* is the qualitative result of the test, that the drugs are associated with the regimen (and not with the viral load test, say, or the patient) and that *after almost 24 weeks* refers to the approximate time of the medical test since the beginning of the regimen.

Queries could also be posed as a series of interrelated questions, each depending on the answers to the previous ones. Answers may depend on results in several resources, such as RxNorm for drug information, the Stanford HIV Drug Resistance Database [1] for clinical cases, and time and date arithmetic for determining the duration of regimens. Investigators would normally select the appropriate sources, specify suitable parameters within the resource's query language, and integrate the answer fragments that are returned into a comprehensive answer. The goal of the Quadri (Question answering about drug resistance information) system is to automate this process by using natural language technology and theorem proving.

1.2 Prior Work

The use of English as a query-interface language has a long history, such as the Precise [2] and Quark [3] systems. Our work is distinguished by (1) using language analysis that does not prematurely eliminate syntactic ambiguity but rather preserves it in a compact form; (2) using domain knowledge to prune ambiguities found during both the language analysis and the search for answers; (3) generating a logical form for a query that captures logical dependencies and that uses a higher level vocabulary interpreted by axioms of the domain theory (handling logical constructs beyond the capabilities of SQL); (4) 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; and (5) giving feedback on a query's logical interpretation and an explanation of how the answer was obtained.

2 Outline of Approach

Our implementation is based on natural language technology (Bridge), which maps English requests for information into queries in a logical language, and reasoning technology (SNARK), which maps those logical queries into invocations of multiple knowledge resources.

2.1 Bridge: Translating English to Logical Form

Bridge [4] is a general natural-language processing system developed at PARC over a two-decade period. It consists of a number of language-processing modules, and can be customized for a specific domain. A posed question is analyzed with a finite-state machine that recognizes named entities and standard English morphology. In our work, we have augmented these entities to included specialized biomedical notations, such as the drug abbreviation symbol *3TC*. The syntactic parser, XLE [5], 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 [6] in which alternative expressions are mapped to a common representation in a knowledge-representation language [7].

Relations among terms are captured by an ontology with domain-specific synonyms and sort structure augmenting a broad-based English lexicon (WordNet). Questions that are ambiguous syntactically may be clear when the subject domain is understood. In the example query, the phrase "*with EFV and 3TC*" can syntactically modify *viral load* or *regimen*. However, *EFV* and *3TC* are drug names, not test results, so they must be syntactically linked to *regimen*. Given the appropriate knowledge, the Bridge system resolves such ambiguities and is able to generate a logical form for the example query. This result is a conjunction of several conditions, such as **patient-has-regimen(?patient, ?regimen)** and **regimen-has-drug-set(?regimen, set(efv, 3tc)).** Logically, 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. 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)),

That is, 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**. To provide user feedback, the constructed logical form is rephrased in 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.

2.2 SNARK: Theorem Proving

The deductive component of Quadri consists of the theorem-proving system SNARK with an appropriate axiomatic theory of HIV drug resistance that we have implemented. SNARK [8] is a first-order-logic theorem prover developed at SRI. It contains many successful inference mechanisms for general-purpose automated reasoning (such as sorted resolution, paramodulation, and rewriting) plus procedures that perform accelerated special-purpose inference (such as 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.

The logical form resulting from Bridge 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 corresponding precise numerical measurement (e.g., 4) 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.

When SNARK is able to prove the theorem using facts collected through procedural attachments, the answer-extraction mechanism yields the patients that satisfy the question and the explanation mechanism produces sentences, similar to the ones 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. Because of the heterogeneity of the 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 service is invoked, by the same procedural-attachment mechanism.

2.3 Query Streams and Anaphoric Reference

Our previous example deals with a single question. The user may pose instead a stream of related queries. At first, she requests *Find patients who were on a regimen*

containing EFV. Seeing the data, she might then ask *Which of those had a high viral load after twenty-four weeks?* Extending the system to support such follow-up questions is our next step. To do so, Quadri must understand that *those* in the second query refers to the patients, not the regimens and not EFV, because patients can have high viral loads, but regimens and drugs cannot.

3 Implementation Status

The Quadri prototype is able to produce the representations shown in this paper. 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 enabling users to provide feedback to distinguish between alternative interpretations and dealing with follow-up questions. Even in its present form, our clinical collaborators have been impressed by Quadri's ability to handle complex ambiguous constructions.

Acknowledgments
Dr. Robert Shafer and Soo-Yon Rhee provided their expertise on the domain and 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 partially funded by grant RC1LM010583 from the National Institutes of Health. 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

1. Rhee, S. Y., et al. Human immunodeficiency virus reverse transcriptase and protease sequence database. *Nucleic Acids Res* 31:298-303, 2003.
2. Popescu, A. M. Etzionie, O., Kautz, H. Towards a Theory of Natural Language Interfaces to Databases. *Proceeding of* IUI 2003.
3. Waldinger R. et al. Deductive Question Answering from Multiple Resources. *New Directions in Question Answering,* AAAI, 2004.
4. Bobrow, D. G. et al, PARC's Bridge and Question Answering System, *Proceedings of the Grammar Engineering Across Frameworks (GEAF07) Workshop*, pp. 46-66, CSLI Publications, 2007.
5. Maxwell, J. T., and Kaplan, R. M. *An efficient parser for* LFG. Butt, M. and King., T. H. (editors), *On-line Proceedings of the LF G96 Conference*. http://csli-publications.stanford.edu/publications. 1996.
6. Crouch, D. and King, T. H. Semantics via F-Structure Rewriting**.** *Proceedings of LFG06*, CSLI On-line publications, 2006.
7. Bobrow D.G., et al. A Basic Logic for Textual Inference. *AAAI Workshop on Inference for Textual Question Answering*, Pittsburgh, PA, 2005.
8. Stickel, M, et al. *A Guide to SNARK*. www.ai.sri.com/snark/tutorial.html.