On the Relation Between "Semantically Tractable" Queries and AURA's Question Formulation Facility

Working Note 34

Peter Clark, peter.e.clark@boeing.com, Sept 2009

1. Introduction

In 2003, Popescu, Etzioni and Kautz published an interesting paper on NL interfaces to databases. They gave a formal characterization of one class of ways that certain NL queries can understood, and thus characterize that class of "straightforwardly understandable" (or, in their terms, "tractable") queries.

In our work on AURA, we are dealing with (controlled) natural language queries to a scientific knowledge base, and this paper explores the extent to which Popescu et al's framework can be applied in the AURA context. The AURA task differs from the database task in several important ways, in particular the variety and complexity of questions is much more open-ended than in Popescu et als database work (Examples of some of the biology questions that AURA is intended to field are discussed in [3]). We briefly summarize Popescu et al's work, and then explore the opportunities and challenges of applying it in a different context.

2. Popescu et al's work

Ana-Maria Popescu et al. [1] present an interesting perspective on natural language (NL) queries to databases, by defining a subclass of queries that can be interpreted straightforwardly (or in their terminology, "tractably"[1]). In other words, of the (potentially infinite) possible interpretations of an NL query, they identify a class of interpretations which are "straightforward and valid" ("tractable"). A NL query with at least one tractable interpretation is said to be a tractable query. They specify a "straightforward and valid" ("tractable") interpretation as follows:

- All the words (bar prepositions and determiners) are mapped to the databases' tables, attributes (column names), or values (for a cell), i.e., there are no unrecognized words in the query.
- Each word interpreted as an attribute (database column) must be syntactically attached to a word interpreted as a valid value (column value) of that attribute (column).
- Each word interpreted as a table (relation) must be syntactically attached to a word interpreted as either a valid attribute (column) or value for that table.

As a word may have multiple, alternative interpretations, and may be attached to more than one other syntactic element, a search algorithm is needed to find an interpretation satisfying the above constraints. It is possible for a query to have zero, one, or many "straightforward and valid" ("tractable") interpretations satisfying the above constraints.

Strictly speaking, the above specification does not fully specify what the interpretation is, it merely specifies constraints on the interpretation. However, the authors implicitly assume the "obvious" interpretation when the above constraints are met, namely:

[1] Their use of the word "tractable" is somewhat misleading, as it has nothing to do with computational complexity.

- a value syntactically attached to an attribute means that (in the final database query) the value = the value of that attribute
- a value not syntactically attached to any attribute will have an "implicit" valid attribute assigned to it by the query interpreter, namely an attribute for which the value is a legal value.

To an approximation, a "tractable" query is one whose words are all within the database's lexicon, and whose syntactic structure mirrors the semantic structure of the database. For queries to a sample of fairly small databases, Popescu et al. found the majority (80%) of queries were "tractable" in this sense.

In a later paper [2], Popescu et al. suggest that the notion of "tractability" can be used to guide parsing, by steering the parser towards parses that have tractable interpretations.

3. Application to AURA

To what extent can we apply these ideas in the context of AURA (biology), i.e., beyond the database context? In AURA, questions are formulated in the controlled English language CPL (Computer-Processable Language) [4].

One nice aspect of Popescu et al.'s work is it provides a (somewhat) formal definition a "straightforward and valid" ("tractable") interpretation. We could define a similar notion in AURA: a tractable query would be one whose words all map directly to KB concepts, and whose syntactic structure reflects the allowable semantic structure in the KB. One could argue that AURA's question interpreter CPL should:

- find the/a tractable interpretation, if one exists
- rephrase the query into a tractable one, if a tractable interpretation does not exist

Here are some observations on this idea:

1. Unlike in Popescu's "small database" domains, where users largely kept to the database vocabulary, the original English biology questions almost always go outside AURA's KB vocabulary; i.e., the biology question vocabulary is much more open and presents a more serious challenge than Popescu et al had to deal with. To be more concrete, in Popescu et al's small database domain, about 95% of the words in a query map to database terms, and at least 80% of their queries are completely free of unrecognized words. In contrast, for 22 original English biology questions (queries) aimed at AURA, only about 60% of the words in a query map to AURA concepts, and only 10% of the queries are completely free of unrecognized words.

   In other words, the large majority of the original English biology questions are "intractable" in Popescu's sense simply because of the more open vocabulary.

   However, we could generalize the definition of tractability to be more accomodating:

   a. rather than insisting that words map directly to KB concepts, we can also allow them to also map indirectly via WordNet (as is currently done in CPL). This raises the number of recognized words (to about 85%, though still only 10% of the queries are completely free of unrecognized words), but introduces the risk of overgeneralization and misinterpretation.

   b. we can generalize Popescu's notion of "syntactic attachment". Popescu insists that words denoting semantically connected concepts are directly syntactically attached

(with a preposition, modifier, or verb-argument relationship). This notion could be generalized to include phrasal attachment also, e.g., "X consists of Y" or "X can be found in Y" could also be considered syntactic attachment of X and Y.

However, not all phrases between objects X and Y suggest a direct semantic relationship – sometimes the phrase describes explicit intermediate objects or events (e.g., “X catalyzes a reaction producing Y”). So care needs to be taken.

2. Intractable (semantically invalid) queries, i.e., with conceptual errors with respect to the KB ontology, are more likely in biology than in the "small database" domain. For example, q9 states

   Q9. Which of the following substances....
   (a) Adenine...

   This is incompatible with AURA's Bio KB, because in the KB Adenine is represented as a Molecule, not a Substance. Such queries are considered intractable in Popescu's framework.

3. It is important to note that a tractable interpretation is not necessarily the right interpretation. This is best understood by remembering that there may be several, distinct tractable interpretations for a given query, only one of which will be the one the user intended. In Popescu et al's "small database" domains, the domains are sufficiently constrained that typically one would not expect more than one tractable interpretation. However, in the AURA domain there can be many. For example, consider an AURA question about

   "…the cytoplasm of a cell…"

   Here, "of" could be interpreted to mean part-of(x,y), or made-of(x,y), or is-inside(x,y). All three alternative interpretations are tractable (in the Popescu sense). However, only one is correct and will lead to the right answer.

   In other words, the notion of "tractable" is necessary but not sufficient to define the desired, baseline behavior of the CPL interpreter.

   Again, it might be possible to extend the notion of tractability for AURA. For example, as well as a “tractable” (semantically valid) interpretation, we might add the notion of a “supported” interpretation that does not just satisfy KB constraints (e.g., domain and range), but also has supporting evidence in the KB. Again consider interpreting “the cytoplasm of the cell”, where “of” can be interpreted as part-of(x,y), made-of(x,y), or is-inside(x,y), but also note that somewhere in the KB there is a statement that “Cytoplasm is part of a Cell”. Although all three interpretations of “of” are tractable (valid), only one is supported and so should be preferred.

4. Popescu et al's notion of tractability concerns validity, but not "model compatibility". That is, we can write queries that (have interpretations that) are tractable (i.e., don't violate any KB constraints, e.g., domain and range), but still are "conceptually mismatched" with respect to the KB, i.e., presuppose a different underlying model.

   For example, a biology question may ask

   "Do peroxisomes catalyze the synthesis of H2O2?"

   while the KB states

   "Peroxisomes synthesize H2O2"

Strictly, the answer to the original question is "no". However, in this case the KFE has encoded a slightly different view of the biology process than that assumed in the question. Note that the problem is not that the question is semantically invalid (it would still be considered "tractable" in Popescu's framework), rather there is a modeling difference between the question and KB. Ideally the QA system would recognize this and suggest a modified question which would be answered. Defining, recognizing, and dealing with such modeling differences is outside the scope of Popescu's framework. It may be something that we should consider addressing in the QF redesign; in particular because in many cases during the "difficult question" analysis the KB representations turned out to be only approximations of those assumed in the questions.

5. CPL's interpreter can sometimes fail to find a "tractable" interpretation even when one exists. This happens when CPL makes a premature, wrong commitment, and is a (minor) deficiency of its current implementation.

The above discussion thus raises the following challenges (each corresponds to the respective point above):

1. How to generalize the specification of "tractability" for the purposes of AURA.
2. How to deal with intractable queries.
3. How to find the intended interpretation from a set of alternative, tractable interpretations.
4. How to deal with tractable queries where modeling differences still exist between the query and KB.
5. How to ensure tractable interpretations are found if they exist.

4. Summary

Popescu's framework is a refreshing approach to clearly specifying a language interpreter's behavior, and offers two key methodological insights:

a. separating the specification of the target behavior from the implementation. This of course is standard practice in many areas of software engineering, but is unusual to see in natural language understanding. It is of course very desirable as it then means that there is a criterion for correctness of implementation, and for exploring different implementation options.

b. realizing that the specification doesn't need to cover all possible inputs. Rather, it is defined for the subset of statements where straightforward NL interpretation is possible.

To an approximation, a "tractable" query in Popescu's framework is one whose words are all within the DB/KB's lexicon, and whose syntactic structure mirrors the semantic structure of the DB/KB. This is a useful notion for defining AURA's behavior, but doesn't quite go far enough. In particular:

a. the specification isn't constrained enough for AURA (there are too many "tractable" interpretations of some queries to AURA, only one of which is right).

b. It doesn't cover enough cases (most AURA queries are "intractable", in particular because of the open vocabulary)

CPL already largely finds tractable interpretations when they exist. Rather, the main problems CPL currently faces are points 2, 3, and 4 discussed in the previous Section, namely (2) dealing with intractable queries (3) dealing with multiple, tractable interpretations, and (4) dealing with tractable interpretations where modeling differences still exist between the query and KB. However, although Popescu et al's notion of tractability does not quite far enough for our

purposes, it provides a refreshing approach to clearly specifying an interpreter's behavior, and one on which we can likely build. In particular, two ways the framework could be extended are:

a. adding in a notion of preference to augment tractability (akin to the preference notions in the database/query relaxation literature), to specify preferred, tractable interpretations.

b. generalizing the specification of tractability to cover more queries, e.g., generalizing the notions of word2concept mapping and syntactic attachment.

These would be interesting future avenues to explore.

References

[1] Popescu, A-M., Etzioni, O., Kautz, H. Towards a Theory of Natural Language Interfaces to Databases. In *Proc IUI'03*, 2003.

[2] Popescu, A-M., Armanasu, A., Etzioni, O., Ko, D., Yates, A. Modern Natural Language Interfaces to Databases: Composing Statistical Parsing with Semantic Tractability. In *Proc. COLING'04*, 2004.

[3] Clark, P. A Study of Some “Hard to Formulate Questions in Biology. Working Note 33, 2009, http://www.cs.utexas.edu/users/pclark/working_notes/

[4] P. Clark, J. Chaw, K. Barker, V. Chaudhri, P. Harrison, J. Fan, B. John, B. Porter, A. Spaulding, J. Thompson, P. Yeh. Capturing and Answering Questions Posed to a Knowledge-Based System. In: Proc 4th Int Conf on Knowledge Capture (KCap'07), 2007.