Knowledge Tracking: Answering Implicit Questions

Reinhard Stolle, Daniel G. Bobrow,
Cleo Condoravdi, Richard Crouch, Valeria de Paiva

Palo Alto Research Center
3333 Coyote Hill Road
Palo Alto, California 94304
{stolle, bobrow, condorav, crouch, paiva}@parc.com

Abstract

Research on Question Answering has produced an arsenal of useful techniques for detecting answers that are explicitly present in the text of a collection of documents. To move beyond current capabilities, effort must be directed toward analyzing the source documents and interpreting them by representing their content, abstracting away from the particular linguistic expressions used. The content representations enable reasoning based on what things mean rather than how they are phrased. Mapping accurately from natural language text to content representations requires deep linguistic analysis and proper treatment of ambiguity and contexts. Research in Question Answering has traditionally tried to circumvent these problems due to the lack of feasible solutions. We strongly believe that these problems can and must be tackled now: PARC's deep NLP technology scales well, and our preliminary results with mapping to content representation are encouraging. In order to bring fundamental issues of deep analysis to the fore, we have chosen to work on a task we call "knowledge tracking" that cannot be accomplished without interpretation of the source text. The goal of knowledge tracking is to identify the relationship of the content of a new document to the content of previously collected documents. Knowledge tracking can thus be viewed as a particular kind of question answering for a set of implicit questions. It provides useful functionality even when applied to a medium-size collection and can therefore serve as a laboratory where deep processing is feasible. Results on this task can help to extend the capabilities of many question-answering systems.

Introduction

Question Answering research has produced a rich set of useful techniques to retrieve possible answers to a user's question from large collections of text documents. Questioners start with an explicit formulation of the desired information, and a document collection that might contain the answer. Often, the document collection has been collected for the purpose of helping in a particular task and is intended for non-casual users. Answering a question often works by using topic-based statistical information retrieval techniques to identify a more limited subset of documents or document fragments that may contain an answer to the question. Natural language techniques are then used to localize further the text fragments that may contain the answer.

At the core of the natural language processing for question answering are techniques that estimate what kind of information the question is asking for (e.g., who, what, where, how much, etc.), and techniques that identify pieces of text—mainly noun phrases—that potentially contain that kind of information. The simplest cases are those of answers matching the linguistic form of the question. Extensions include ad-hoc transformations among sets of related expressions, connecting, for example, "buy" and "sell," "kill" and "die."

In our research, we are working on "knowledge tracking." This task is closely related to question answering, but it requires processing that goes beyond detection of relevant text fragments. Knowledge tracking monitors the content of an incoming stream of natural language documents and identifies their relation to those already in the collection. Knowledge tracking can be viewed as a particular kind of question answering, namely answering implicit questions such as "what's new," "what's interesting."

This task requires analyzing and interpreting the source documents by representing their content. Analyzing the source documents necessitates deep and efficient syntactic analysis, capable of dealing with long sentences (30 or more words). Deep syntactic analysis must be coupled with a way of managing ambiguity, rife in natural language texts. Representing content requires mechanisms for detecting and appropriately representing the context in which a statement is made, such as conditional options, beliefs, and counterfactual suppositions. These difficult issues have been recognized as central to explicit question answering as well (e.g., Burger et al 2002). For example, answers are often neither local in the documents nor explicitly expressed in the text. However, conventional wisdom holds that these issues need to be sidestepped or worked around for practical purposes. Contradictory to this widespread belief, recent results (Crouch et al., 2002; Riezler et al., 2002) show that deep symbolic natural language processing and mapping to content representation is feasible for appropriate problems, such as knowledge tracking. The purpose of this short position paper is to outline our approach and illustrate its importance with an example. Technical details have been published elsewhere.

In the next section, we formulate the knowledge tracking problem and discuss some of the underlying capabilities that are needed to approach a solution. We believe that these same language, context, and ambiguity management capabilities will be necessary for the research programme outlined in the Q&A Roadmap Paper (Burger et al., 2002). We will conclude with a brief description of a prototype system for knowledge tracking that demonstrates progress toward the capabilities to answer some implicit questions, and that incorporates early versions of these capabilities.

Knowledge Tracking and Implicit Questions

We have begun to develop techniques that enable the tracking of content change in growing collections of natural language documents. Knowledge tracking can be a high value task for a community that must monitor a growing collection of texts that contain important information with respect to a task they are engaged in. Given a collection of previously seen text documents, knowledge tracking monitors an incoming stream of documents, identifying how the content of new documents relates to the content of the collection of previously seen documents. Two content relations are central: *contradiction*—part of one document contradicts (is inconsistent with) part of another, and *redundancy*—part of one document conveys the same content as (or is entailed by) part of another. Once these central relations can be identified they provide a basis for identifying more complex content relations such as elaboration (redundancy plus linked additional information), presupposition, or consequence. It is important to note that entailment and contradiction are *logical* content relations that are not captured by traditional topic-based techniques for similarity detection (e.g., Wayne, 2000; Hatzivassiloglou et al., 1999; Brants and Stolle, 2002). An ability to recognize such logical relationships allows users of document collections to be notified of incoming documents that relate to previously seen documents in a way relevant to the user's task at hand.

To make this more concrete, consider a scenario associated with a knowledge base being shared by service technicians for photocopiers and printers at Xerox. Service technicians often run into difficult problems and want to make use of the experience of other technicians. Many technicians share their experience by writing tips, in free text, detailing difficult problems they have encountered in the field, and possible plans of action for handling them. Technicians facing a broken machine can search this knowledge base for relevant documents to give them clues about how to proceed. The database has an index that allows rapid retrieval of documents containing query terms. Implicitly, technicians are looking for connections between the symptoms they are seeing, and possible faults, and/or fixes. They scan the retrieved tips very rapidly and often find something that helps. Users say that using this high quality knowledge base is very much like having a conversation with other technicians.

Maintaining the quality of this knowledge base is a prime example of a knowledge tracking application. Once the collection grows beyond a certain size, an issue of critical importance is how a human can monitor new inputs without having to (re)read a large number of earlier reports. Implicitly each new document poses a number of questions. First, is this a solution to a problem that has previously been seen? If so, is all the information in the new tip contained in earlier tips? A redundant tip should not be added to the database—technicians complain when queries get redundant answers. If not, what is the relationship of the new information to what is already there? Is it an elaboration of the problem, or of the solution? Is it an alternative solution—if so, under what conditions should it be used? Is it contradictory to what was written before? Was the previous statement correct at the time it was entered, and this is just an evolution over time? A knowledge tracking software agent that can suggest to a human monitor potential relations between a new document and previously seen documents would be useful in many contexts.

As an example consider the two reports below from that collection of tips. Suppose R118 has just arrived as a new report.

R57:	R118:
The left handler cover safety cable breaks, allowing the left cover to pivot too far, breaking the cover. Remove the plastic sleeve from around the cable. Cutting the plastic off the cable makes it more flexible, which prevents cable breakage.	The current safety cable used in the handler fails prematurely, causing the left cover to break. The plastic jacket makes the cable too stiff, which causes stress to be concentrated on the cable ends where it eventually breaks. When the old cable fails replace it with the new one, which has a shorter jacket.

These two reports are about the same problem, and they give a similar analysis as to why the problem occurs. However, they have some very interesting differences. R118 elaborates on the cause of the cable breakage (cumulative stress). The two reports suggest different fixes for the situation: whereas R57 suggests a short-term "work-around" fix to the problem, R118 presents the "official" solution. A monitor who sees these two tips may provide some meta-information about their different character (work-around versus official) and link the two (so that, for example, even if sleeve were the content word used to retrieve relevant tips, R57 would "drag in" R118). A technician could then see both the work-around for when a new part were not available, and the official repair.

We have seen implicit questions such as: "What's new compared to what I already know?" "What's new and interesting?" "What's new and contradictory to what I have seen so far?" Being able to sensibly answer these kinds of question requires, of course, good characterizations of "new," "interesting," and "contradictory." In the long run, one can imagine a whole range of heuristics that may provide such definitions, either explicitly given by the user, or based on a model of a user or a user community, taking into account, for example, users' context, history of actions, and so on.

Implicit questions typically arise from tasks that are performed by expert users in focused domains, similar to the "Professional Information Analyst" described in the Q&A Vision Statement (Carbonell et al., 2000). In such "communities of common task," users share the overall task in a common domain; however, their focus of interest may change dynamically. In our repair tips example, the overall task may be the maintenance of the quality of the tip database, and the domain is the repair of photocopier and printer equipment. The interest of a particular user at a certain point in time will likely be more focused. For example, one user may be interested in solutions that involve a specific newly released part; another user may be interested in particular problem-solution relationships, such as a work-around solution for the safety cable problem. Yet another user may focus on contradictory solutions to the same problem. What these examples show is that computational techniques aimed at supporting such tasks must (1) operate on data structures that represent the *content* of the text, rather than on raw linguistic expressions, and (2) be able to compare or otherwise relate pairs of text fragments based on their content and *content structure*.

Our group at PARC is using our knowledge tracking task as a driver to develop and test these capabilities to meet these challenges. These include:

1. The content of natural language text must be *represented* in such a way as to facilitate reasoning about the content. A solution to this problem must allow for *non-local reasoning*: reasoning that involves the content of different text fragments. Furthermore, the reasoning must be able to take advantage of external knowledge sources (e.g., domain knowledge, Cyc). Our mantra is that documents are linguistic expressions of content, and the answers lie in the content and the inferences that can be drawn immediately from that content.
2. The representation must appropriately capture the *contexts* relative to which the represented sets of content are claimed to be true. When determining the relationship between two content fragments, their contexts must be taken into account. For example, two mutually inconsistent statements are not necessarily contradictory if they are embedded in contexts that make inconsistent assumptions. At a high level, different contexts may represent different beliefs, perspectives, temporal states, etc. At a medium level, contexts may represent counterfactual worlds, conditionals, or states in action sequences. At a low level, contexts encode scope of variables and negation.
3. One of the hardest issues with natural language is its inherent *ambiguity*. Resolving ambiguity too early leads to incorrect representations. The algorithms that analyze the text and produce the content representation must represent and manage ambiguity, and disambiguate only when they can do so reliably.

In the next section, we report briefly on *Katie*, a prototype knowledge tracking system we have built to explore how these capabilities can work together for this task.

Introducing Katie

As described in the previous section, a prerequisite for the functionality we hope to achieve for implicit questions is the ability to identify conceptual relationships between the content of two text fragments. This, in turn, requires the representation of the conceptual contents of the texts in a KR formalism. One of the difficult problems in this endeavor is to derive the content structure of the text from its linguistic structure. We then use structure matching to identify content overlaps as a place to look for interesting content relations. Two content relations are central: *Contradiction*—part of one document contradicts part of another (e.g., "replace the cable" vs. "remove its jacket"). *Redundancy*—part of one document conveys the same content as part of another (e.g., "the jacket makes the cable stiff" and "removing the jacket makes the cable more flexible"). Detecting contradiction and redundancy is also key to tracking more complex content relations, such as elaboration, presupposition, or consequence.

We have made significant progress on this problem in a proof of concept system. A deep analysis based on Lexical Functional Grammar theory (Kaplan and Bresnan, 1982) combined with Glue Semantics (Dalrymple, 1999) produces a compact representation of the syntactic and semantic structures for each sentence. From this language-driven representation of the text, we map to a knowledge-driven representation of the contents that abstracts away from the particular natural language expression. This mapping includes several—not necessarily sequential—steps. In one step, we rely on a domain-specific ontology to identify canonicalized entities and events that are talked about in the text. In our case, these entities and events include things like parts, e.g., photoreceptor belt, and relevant activities such as cleaning, for example. Another step performs thematic role assignments and assembles fragments of conceptual structures from the normalized entities and events (e.g., cleaning a photoreceptor belt).

Furthermore, certain relations are normalized; for example, "stiff" and "flexible" in the repair tips example above both refer to the rigidity of an object, one being the inverse of the other; see (Everett et al., 2002).

Yet another step composes structure fragments into higher-level structures that reflect causal or temporal relations, such as action sequences or repair plans. All steps involve ambiguity resolution as a central problem, which requires inference based on extensive linguistic and world knowledge. We are currently investigating how to appropriately represent contexts in the KR and how to derive these KR contexts from the Glue semantics. For a more detailed description of our approach to natural language processing and mapping to KR, and for a description of its scalability, see (Crouch et al., 2002).

Finally, we assess the similarity of two documents by running a variant of the Structure Mapping Engine (SME) (Forbus et al., 1989) on the canonicalized content representations of the documents. SME anchors its matching process in identical elements that occur in the same structural positions in the base and target representations, and from this builds a correspondence. The larger the structure that can be recursively constructed in this manner, while preserving a systematicity constraint of one-to-one correspondence between base and target elements and the identity of anchors, the greater the similarity score. Our goal is to use content overlaps as identified by SME as a starting point for identifying redundancies, inconsistencies and other content relations. Since the content representations are canonicalized, the identification of content relationships will only require lightweight reasoning, such as moving up and down in a concept hierarchy, formalized in description logic. For example, "plug the signal wire into the lower socket" is a specialization of "connect the wire and the socket."

Matching of content structures raises several interesting problems. For example, establishing correspondences between thematic role assignments requires more sophistication than just, say, matching the direct agents and patients of actions, as the following example shows. "Scrape the dirt off the pin" should be recognized as a specialization of "clean the pin," even though the direct patient of the former is the dirt, whereas the direct patient of the latter is the pin. Another interesting problem is the question when to consider statements contradictory: is replacing the safety cable contradictory or redundant to modifying the sleeve of the safety cable? Part of the solution to this problem will be to identify domain and/or task-specific contexts and typical higher-level content structures. For example, whether the statement "the voltage is 7V" is redundant or contradictory to the statement "the voltage is 7.2V" depends on the user's task at hand. One of the challenges for knowledge tracking is to avoid hard-coding such task-specific decisions into the mapping from language to KR or into the structural content matcher.

Conclusion

The issues we are exploring in knowledge tracking, the task of answering implicit questions, are strongly related to the six dimensions of challenges—scope, context, judgement, multiple sources, fusion, and interpretation—laid out in the Q&A Vision Statement (Carbonell et al., 2000) and Roadmap Paper (Burger et al., 2001). We are working with a relatively small, focused document collection, which makes feasible the deep processing of the source documents that we believe is crucial for long term progress in question answering. The identification of the fundamental relations of inconsistency and redundancy are precursors to building more elaborate descriptions of the relationship of new information to old, and to finding non-local relationships that may serve to answer questions more completely from the knowledge in a collection.

References

1. Brants, T., Stolle, R. 2002. Finding Similar Documents in Document Collections. *Proc. LREC-2002 Workshop on Using Semantics for Information Retrieval and Filtering*, Las Palmas, Spain.
2. Burger, J., Cardie, C., Chaudhri, V., Gaizauskas, R., Israel, D., Jacquemin, C., Lin, C.-Y., Maiorano, S., Miller, G., Moldovan, D., Ogden, B., Prager, J., Riloff, E., Singhal, A., Shrihari, R., Strzalkowski, T., Voorhees, E., Weischedel, R. 2001. Issues, tasks, and program structures to roadmap research in question & answering (Q&A). http://www-nlpir.nist.gov/projects/duc/roadmapping.html
3. Carbonell, J., Harman, D., Hovy, E., Maiorano, S., Prange, J., Sparck-Jones, K. 2000. Vision Statement to Guide Research in Question & Answering (Q&A) and Text Summarization. http://www-nlpir.nist.gov/projects/duc/roadmapping.html
4. Crouch, R., Condoravdi, C., Stolle, R., King, T., de Paiva, V., Everett, J.O., Bobrow, D.G. 2002. Scalability of Redundancy Detection in Focused Document Collections. *Proc. SCANALU-2002*.
5. Dalrymple, M., ed. 1999. *Semantics and Syntax in Lexical Functional Grammar: The Resource Logic Approach*. Cambridge, Mass.: MIT Press.
6. Everett, J.O., Bobrow, D.G., Stolle, R., Crouch, R., de Paiva, V., Condoravdi, C., van den Berg, M., Polanyi, L. 2002. Making Ontologies Work for Resolving Redundancies across Documents. *Communications of the ACM* **45**(2):55-60.
7. Forbus, K.D., Falkenhainer, B., Gentner, D. 1989. The structure mapping engine: Algorithm and examples. *Artificial Intelligence* **41**(1):1-63.
8. Hatzivassiloglou, V., Klavans, J., Eskin, E. 1999. Detecting Similarity by Applying Learning over Indicators. *Proc. ACL-1999*.
9. Kaplan, R.M., Bresnan, J. 1982. Lexical-Functional Grammar: A Formal System for Grammatical Representation. In Bresnan, J., ed., *The Mental Representation of Grammatical Relations*. Cambridge, Mass.: MIT Press. Reprinted in Dalrymple, M., Kaplan, R., Maxwell, J., Zaenen, A., eds., Formal Issues in Lexical-Functional Grammar. CSLI 1995.
10. Riezler, S., King, T., Crouch, R., Maxwell, J., Kaplan, R.M., Johnson, M. 2002. Parsing the Wall Street Journal using a Lexical-Functional Grammar and Discriminative Estimation Techniques. *Proc. ACL-2002*. Philadelphia, PA.
11. Wayne, C. 2000. Multilingual Topic Detection and Tracking: Successful Research Enabled by Corpora and Evaluation. *Proc. LREC-2000*.