Question-Answering Using Semantic Relation Triples

Kenneth C. Litkowski
CL Research
9208 Gue Road
Damascus, MD 20872
ken@clres.com
http://www.clres.com

Abstract

This paper describes the development of a prototype system to answer questions by selecting sentences from the documents in which the answers occur. After parsing each sentence in these documents, databases are constructed by extracting relational triples from the parse output. The triples consist of discourse entities, semantic relations, and the governing words to which the entities are bound in the sentence. Database triples are also generated for the questions. Question-answering consists of matching the question database records with the records for the documents.

The prototype system was developed specifically to respond to the TREC-8 Q&A track, with an existing parser and some existing capability for analyzing parse output. The system was designed to investigate the viability of using structural information about the sentences in a document to answer questions. The CL Research system achieved an overall score of 0.281 (i.e., on average, providing a sentence containing a correct answer as the fourth selection). The score demonstrates the viability of the approach. Post-hoc analysis suggests that this score understates the performance of the prototype and estimates that a more accurate score is approximately 0.482. This analysis also suggests several further improvements and the potential for investigating other avenues that make use of semantic networks and computational lexicology.

1. Introduction

CL Research is primarily focused on investigating the manner in which computational lexicons can be used for natural language processing tasks. This research primarily involves the development of methods for constructing computational lexicons (particularly through analysis of machine-readable dictionaries) and examining ways that these lexicons can be used in such tasks as word-sense disambiguation and text summarization.

The CL Research question-answering prototype extended functionality of the DIMAP dictionary creation and maintenance software, which includes some components intended for use as a lexicographer's workstation.[1] The TREC-8 Q&A track provided an opportunity not only for examining use of computational lexicons, but also for their generation as well, since many dictionaries (particularly specialized one) contain encyclopedic information as well as the usual genus-differentiae definitions. The techniques developed for TREC and described herein are now being used for parsing dictionary definitions to help construct computational lexicons that contain more information about semantic relations, which in turn will be useful for natural language processing tasks, including question-answering.

2. Problem Description

Participants in the TREC-8 QA track were provided with 200 unseen questions to be answered from the TREC CD-ROMs, (about 1 gigabyte of compressed data), containing documents from the *Foreign Broadcast Information Service*, *Los Angeles Times*, *Financial Times*, *Congressional Record*, and *Federal Register*. These documents were stored with SGML formatting tags. Participants were given the option of using their own search engine or of using the results of a "generic" search engine. CL Research chose the latter, obtaining 200 megabytes of data, with the top 200 documents retrieved by the search engine. These top documents were provided a couple of weeks before the deadline.

[1]Demonstration and experimental versions of DIMAP are available at http://www.clres.com.

Participants were then required to answer the 200 questions in either 50-byte answers or by providing a sentence or 250-byte string in which the answer was embedded. For each question, participants were to provide 5 answers, with a score attached to each for use in evaluating ties. NIST evaluators then judged whether each answer contained a correct answer. Scores were assigned as the inverse rank. If question q contained a correct answer in rank r, the score received for that answer was 1/r. If none of the 5 submissions contained a correct answer, the score received was 0. The final score was then computed as the average score over the entire set of questions.

In the prototype implementation, CL Research submitted sentences, although for some types of questions, answers were also developed for potential 50-byte submission.

3. System Description

The CL Research prototype system consists of four major components: (1) a sentence splitter that separated the source documents into individual sentences; (2) a parser which took each sentence and parsed it, resulting in a parse tree containing the constituents of the sentence; (3) a parse tree analyzer that identified important elements of the sentence and created semantic relation triples stored in a database; and (4) a question-answering program that (a) parsed the question into the same structure for the documents, except with an unbound variable, and (b) matched the question database records with the document database to answer the question. The matching process first identified candidate sentences from the database, developed a score for each sentence, and chose the top 5 sentences for submission.

3.1 Sentence Identification in Documents

The parser (described more fully in the next section) contains a function to recognize sentence breaks. However, the source documents do not contain crisply drawn paragraphs that could be submitted to this function. Thus, a sentence could be split across several lines in the source document, perhaps with intervening blank lines and SGML formatting codes. As a result, it was first necessary to reconstruct the sentences, interleaving the parser sentence recognizer.

At this stage, we also extracted the document identifier and the document date. Other SGML-tagged fields were not used. The question number, document number, and sentence number provided the unique identifier when questions were answered to extract the appropriate sentence from the document.

For the TREC-8 QA run submitted to NIST, only the top 10 documents (as ranked by the search engine) were analyzed. Overall, this resulted in processing 1977 documents from which 63,118 sentences were identified and presented to the parser. Thus, we used an average of 31.9 sentences per document or 315.5 sentences in attempting to answer each question.

3.2 Parser

The parser used in TREC-8 (provided by Proximity Technology) is a prototype for a grammar checker. The parser uses a context-sensitive, augmented transition network grammar of 350 rules, each consisting of a start state, a condition to be satisfied (either a non-terminal or a lexical category), and an end state. Satisfying a condition may result in an annotation (such as number and case) being added to the growing parse tree. Nodes (and possibly further annotations, such as potential attachment points for prepositional phrases) are added to the parse tree when reaching some end states. The parser is accompanied by an extensible dictionary containing the parts of speech (and frequently other information) associated with each lexical entry. The dictionary information allows for the recognition of phrases (as single entities) and uses 36 different verb government patterns to create dynamic parsing goals and to recognize particles and idioms associated with the verbs (the context-sensitive portion of the parser).

The parser output consists of bracketed parse trees, with leaf nodes describing the part of speech and lexical entry for each sentence word. Annotations, such as number and tense information, may be included at any node. The parser does not always produce a correct parse, but is very robust since the parse tree is constructed bottom-up from the leaf nodes, making it possible to examine the local context of a word even when the parse is incorrect. In TREC-8, the parse output was unusable for only 526 of the 63,118 sentences (0.8 percent). Usable output was available despite the fact that there was at least one word unknown to the parsing dictionary in 5,027 sentences (8.0 percent).

3.3 Document and Question Database Development

The key step in the CL Research question-answering prototype was the analysis of the parse trees to extract semantic relation triples and populate the databases used to answer the question. A **semantic relation triple** consists of a discourse entity, a semantic relation which characterizes the entity's role in the sentence, and a governing word to which the entity stands in the semantic relation.

In general terms, the CL Research system is intended to be part of a larger discourse analysis processing system (Litkowski & Harris, 1997). The most significant part of this system is a lexical cohesion module intended to explore the observation that, even within short texts of 2 or 3 sentences, the words induce a reduced ontology (i.e., a circumscribed portion of a semantic network such as WordNet (Fellbaum, 1998) or MindNet (Richardson, 1997)). The objective is to tie together the elements of a discourse (in this case, a document) using lexical chains and coreference to create a hierarchical characterization of a document. The implementation in TREC-8 does not attain this objective, but does provide insights for further development of a lexical cohesion module.

The first step of this discourse processing is the identification of suitable discourse entities. For TREC-8, this involved analyzing the parse tree **node** to extract numbers, adjective sequences, possessives, leading noun sequences, ordinals, time phrases, predicative adjective phrases, conjuncts, and noun constituents as discourse entities . To a large extent, these entities include, as subsets, named entities and time expressions as single entities (although not specifically identified as such in the databases).

The semantic relations in which entities participate are intended to capture the semantic roles of the entities, as generally understood in linguistics. This includes such roles as agent, theme, location, manner, modifier, purpose, and time. For TREC-8, we did not fully characterize the entities in these terms, but generally used surrogate place holders. These included “SUBJ,” “OBJ”, “TIME,” “NUM,” “ADJMOD,” and the prepositions heading prepositional phrases.

The governing word was generally the word in the sentence that the discourse entity stood in relation to. For “SUBJ,” “OBJ,” and “TIME,” this was generally the main verb of the sentence. For prepositions, the governing word was generally the noun or verb that the prepositional phrase modified. (Because of the context-sensitive dynamic parsing goals that were added when a verb or a governing noun was recognized, it was possible to identify what was modified.) For the adjectives and numbers, the governing word was generally the noun that was modified.

The semantic relation and the governing word were not identified for all discourse entities, but a record for each entity was still added to the database for the sentence. Overall, 467,889 semantic relation triples were created in parsing the 63,118 sentences, an average of 7.4 triples per sentence.

The same functionality was used to create database records for the 200 questions. The same parse tree analysis was performed to create a set of records for each question. The only difference is that one semantic relation triple for the question contained an unbound variable as a discourse entity. The question database contained 891 triples (for 196 questions), an average of 4.5 triples per question.

3.4 Question Answering Routines

For TREC-8, a database of documents was created for each question, as provided by the NIST generic search engine. A single database was created for the questions themselves. The question-answering consisted of matching the database records for an individual question against the database of documents for that question.

The question-answering phase consists of three main steps: (1) coarse filtering of the records in the database to select potential sentences, (2) more refined filtering of the sentences according to the type of question, and (3) scoring the remaining sentences based on matches between the question and sentence database records. The sentence were then ordered by decreasing score for creation of the answer file submitted to NIST.

3.4.1 Coarse Filtering of Sentences

The first step in the question-answering phase was the development of an initial set of sentences. The discourse entities in the question records were used to filter the records in the document database. Since a discourse entity in a record could be a multiword unit (MWU), the initial filtering used all the individual words in the MWU. The question and sentence discourse entities were generally reduced to their root form, so that issues of tense and number were eliminated. In addition, all words were reduced to lowercase, so that issues of case did not come into play during this filtering step. Finally, it was not necessary for the discourse entity in the sentence database to have a whole word matching a string from the question database. Thus, in this step, all records were selected from the document database having a discourse entity that contained a substring that was a word in the question discourse entities.

The join between the question and document databases produced an initial set of unique (document number, sentence number) pairs that were passed to the next step.

3.4.2 Refinement of Viable Sentences

The second step of the question-answering process applied more detailed screening of the sentences. This screening involved the application of criteria based on the type of question.

As indicated above, one record associated with each question contained an unbound variable as a discourse entity. The type of variable was identified when the question was parsed and this variable was used to determine which type of processing was to be performed during the sentence refinement step.

The prototype system recognized six question types (usually with typical question elements): (1) **time** questions ("when"), (2) **location** questions ("where"), (3) **who** questions ("who" or "whose"), (4) **what** questions ("what" or "which," used alone or as question determiners), (5) **size** questions ("how" followed by an adjective), and (6) **number** questions ("how many"). Question phraseology not envisioned during the prototype development (principally questions beginning with "why" or non-questions beginning with "name the ...") were assigned to the **what** category, so that question elements would be present for each question.

Some adjustments to the question type were made just prior to the refined filtering. Specifically, it was recognized that questions like "what was the year" or "what was the date" and "what was the number" were not **what** questions, but rather **time** or **number** questions. Other phraseological variations of questions are likely and could be made at this stage.

In general, the functionality for the screening step involved elimination of sentences from further processing (based on criteria described below) and initialization of the data structure for holding a 50-byte answer. An initial score (of 1000) was assigned for each sentence during this process. And, the number of viable sentences was limited.

<u>1. Time Questions</u> - The first criterion applied to a sentence was whether it contained a record that has a TIME semantic relation. The parser has specific mechanisms for recognizing prepositional phrases of time or other temporal expressions (e.g., "last Thursday"). During the analysis of the parser output, the database records created for these expressions were given a TIME semantic relation. After screening the database for such records, the discourse entity of such a record was then examined further. If the discourse entity contained an integer or any of its words were marked in the parser's dictionary as representing a time period, measurement time, month, or weekday, the discourse entity was selected as a potential answer.

<u>2. Where Questions</u> - Each sentence was examined for the presence of "in" as a semantic relation. The discourse entity for that record was selected as a potential answer.

<u>3. Who Questions</u> - There was no elimination of sentences for these questions. All sentences were continued to the next step. A potential answer was developed by searching for a record that had the same governing word as that of the unbound variable. (For example, "who created ..." would show "create" as the governing word; a match would be sought for a sentence record with "create" as the governing word.) The head noun of the discourse entity would be the potential answer.

<u>4. What Questions</u> - There was no elimination of sentences for these questions. All sentences were continued to the next step. A potential answer was developed by searching for a record that had the same governing word as that of the unbound variable. The discourse entity would be the potential answer.

<u>5. Size Questions</u> - The first criterion applied to a sentence was whether it contained a record that has a NUM semantic relation. The parser has specific mechanisms for recognizing numbers. During the analysis of the parser output, the database records created for these expressions were given a NUM semantic relation. If these expressions were followed by a noun, the noun would be captured as the

governing word. After screening the database for NUM records, the governing word of such a record was then examined further. If any of the words of the discourse entity were marked in the parser's dictionary as representing a measure, a unit, or a measurement size, the discourse entity, a space, and the governing word were constructed as a potential answer.

6. Number Questions - The same criterion as used in size questions was applied to a sentence to see whether it contained a record that has a NUM semantic relation. In these cases, the number itself (the discourse entity) was selected as the potential answer.

3.4.3 Sentence Scoring

Each sentence that passed the screening process of the previous step was assigned a base score of 1000 and was then evaluated for further correspondences to the question database records. Each record of the question database was examined in relation to each record for the sentence in the document database. Points were added or deducted based on correspondences.

If the discourse entity in the sentence record is a proper or complete substring of the discourse entity in the question record, 5 points are added when the semantic relation or governing word match completely. Five points are deducted if the match is not complete.

If the question discourse entity is an MWU, each word of the MWU is examined against the discourse entity in the sentence record. If a word in the MWU is a substring of the sentence discourse entity, 5 points are added to the score. If the last word of the MWU is a substring of the sentence discourse entity (generally corresponding to the head noun of the MWU), 20 points are added. When we have a substring match, we also test the semantic relation and the governing word of the two records, adding 5 points for each match.

In general, then, points are added because of matches in the semantic relation and governing word fields, but only when there is at least a partial match between the discourse entities of the two records. Thus, the focus of the matching is on the structural similarity between the question records and the sentence records, i.e., on whether the discourse entities participate in the same type of semantic relations with the same governing word. Many of the sentences passed to this step will have minimal changes to their scores, while those that match on structural similarity will tend to separate out relative to other sentences in the documents.

After scores have been computed for all sentences submitted to this step, the sentences are sorted on decreasing score. Finally, the output is constructed in the desired format (for both 50-byte and 250-byte answers), with the original sentences retrieved from the documents.

4. TREC-8 Q&A Results

The official score for the CL Research 250-byte sentence submission was 0.281. This means that, over all questions, the CL Research prototype provided a sentence with the correct answer as its 4th choice. This compares to an average score of 0.332 among the 25 submissions for the TREC-8 Q&A track (i.e., a correct answer in the 3rd position). In examining all the answers submitted by the various teams, the CL Research prototype was one of only two teams that submitted full sentences, as opposed to a 250-byte window around an answer.

The CL Research prototype submitted sentences containing correct answers for 83 of the 198 questions. Compared to the median scores for the 198 questions, the CL Research prototype was better than the median for 40 questions, equal for 109 questions, and less for 49 questions. Since CL Research did not provide a correct sentence for 115 questions, this means that for 66 questions, the median score among the 25 participating systems was unable to provide a correct answer. Finally, the CL Research prototype equaled the best score for 46 questions and the worst score for 115 questions (i.e., the questions where CL Research did not provide a correct answer).

The CL Research prototype performed better than the average score of 0.332 for 56 questions. On these questions, the average score was 0.447 (that is, a correct answer was given as the 2nd ranked answer over all participating systems). Thus, the questions for which the CL Research prototype provided correct answers were in general easier than the remaining questions. However, among these questions, 39 were easier than the average and 17 were more difficult than the average of 0.332. In other words, the CL Research prototype did not just answer the easier questions, but was able to answer some of the more difficult questions as well.

5. Analysis

The results achieved by the CL Research prototype seem to indicate that the general approach of matching relational structures between the questions and the documents is viable. The prototype selected 937 sentences and at least 83 correct sentences out of over 63,000 sentences in the source documents, so the approach clearly performed much better than random. Since the prototype was an initial implementation, focused primarily on just providing a set of answers, without any evaluation of alternative approaches and without inclusion of several generally available research findings (e.g., named-entity recognition, time phrase computations, and coreference resolution), the approach seems to have much promise.

Even with the claimed level of performance, however, it seems that the official results significantly understate the viability of the general approach in the prototype. This statement is based primarily on the fact that only the top 10 documents were used in an attempt to answer the questions, when frequently an answer did not appear in any of these documents. There are several other simple changes (such as resolution of relative time phrases to a specific date, where the appropriate phrase was identified in the prototype as one of the submitted answers) which would result in a higher score.

Overall, based on post-hoc analysis of the cases where the CL Research prototype did not provide the correct answer, it is estimated that a more accurate overall score is approximately 0.482. (This estimate is based on post-hoc analysis of 25 percent of the questions where no correct answer was provided.) The reasons justifying this estimate are detailed below.

<u>1. Cutting off sentences</u> - For three questions, the limitation to 250-byte strings cut off the portion that would have recognized by NIST evaluators as correct. In each case, the appropriate sentence was ranked first, adding 0.015 to the overall score.

<u>2. Inclusion of document containing answer</u> - Post-hoc analysis revealed that for one-third of the questions, the answer was not in the top 10 documents included in the database for the question. When a document containing the answer was added to the database, correct answers were identified in two-thirds of the cases, with an average inverse rank of 0.320, adding 0.153 to the overall score.

<u>3. Relative time resolution</u> - One-fifth of the questions answered incorrectly required resolution of relative time phrases ("last Thursday," "today," "two years ago"). The functionality for this time resolution is essentially already present in the CL Research prototype and the document date necessary for this computation is contained in the document databases. The average inverse rank for the sentences provided in the prototype results is 0.292, adding 0.033 to the overall score.

There are some considerations in addition to the above that also would portray the CL Research prototype more favorably, but for which no immediate estimate of improvement in the overall score is claimed. For 6 percent of the incorrect answers, a sentence containing the correct answer was generated and was tied with an answer that was submitted. However, because the sentence was not generated in a timely order, it was not submitted.

The correct answer was also generated for another 20 percent of the cases where no correct answer was provided, but the appropriate sentences were ranked lower than those submitted. In another 17 percent of the cases with no correct answers, the answer required coreference resolution from one of the sentences submitted. About 10 percent of the cases involved incomplete creation of document database entries due to bugs in the parser or in the mechanisms for extracting database records; it is not yet clear what effect correcting these difficulties would have. (These difficulties resulted in no sentences being submitted for 6 questions.)

Further examination of the results is necessary to understand the factors contributing to success and failure. A first analysis investigated the relation of the question parsing and database construction to whether a question was answered correctly. About 65 percent of the correct answers had no problems with parsing and database construction for the question, compared to 51 percent of questions with incorrect answers. Only 20 percent of the questions with correct answers had parsing problems, compared to 25 percent for those with incorrect answers. About 8 percent of the questions with correct answers involved questions not overtly recognized by the prototype (where a default **what** question type was used), compared to 11 percent of the questions with incorrect answers. Finally, 7.5 of the questions with correct answers had words unknown to the parsing dictionary, compared to 4.5 percent for questions with incorrect answers.

There were no obvious correlates for the scores. For questions with correct answers, the scores were tabulated by the number of database records generated by the questions (which ranged from 2 to 10, with an average of 4.4 records). Higher scores would have been expected for those questions with a higher number of records, but instead the average scores were about the same across the range. Similarly, the average scores for the top 5 rank answers were nearly identical for questions answered correctly and questions answered incorrectly. The average scores for the submitted answers were weakly correlated with the difficulty of the questions (0.18) reported by NIST. However, the correlation was lower (0.11) for questions answered correctly and much higher (0.25) for the questions answered incorrectly. This result is somewhat paradoxical.

Further detailed analysis is necessary to get at the most significant contributors to the scores. Heuristics were used in developing the scoring mechanisms. At this time, these heuristics have not been evaluated, either for identifying the valid or invalid contributions to the scores or for evaluating the weighting scheme.

6. Anticipated Improvements

The immediate possibilities for improvements are many and the possibilities for exploration are quite diverse. In addition, there are opportunities to be explored for integrating the prototype within more generalized search engines. Finally, the prototype can be examined for suitability for use with specialized textual databases.

The immediate improvements are evident from the analysis indicated above: (1) dealing with problem cases where answers weren't generated because of problems with parsing the questions or extracting appropriate database triples from the questions; (2) addition of time phrase analysis routines; (3) extension of the question types handled by the system; and (4) problems in extracting the document database triples arising from the parser or extraction routines.

Less immediate, but straightforward extensions can be gained by incorporating (1) coreference resolution techniques and (2) named entity techniques. The database extraction routines constituted a minimal implementation. These can easily be extended by further analysis of the parse output.

At the next level of complexity, the database extraction techniques require further refinement and extension. The semantic relations used in the prototype can be enhanced, particularly beyond the use of specific prepositions as the characterizing element. The reliance on specific prepositions is likely to have reduced matches; generalizing specific prepositions to more general semantic relations would yield better matches that would not be dependent on specific phraseology. The next step along these lines would involve incorporation of better discourse analysis techniques (such as text summarization research (Mani & Bloedorn, 1999)) for tying together records in the document databases.

Along the same lines, the prototype could be improved by incorporating techniques from lexical cohesion and lexical chain research to tie database records together (Green, 1997). This would specifically involve use of semantic network data (such as is present in WordNet or MindNet), particularly to link synonymic and hypernymic phraseology. Larger dictionaries would also be of some use.

Finally, the mechanisms in the prototype can be improved. Further post-hoc analysis will likely lead to better analysis and selection of sentences. The mechanisms for examining the selected sentences (during the analysis of specific question types) were also somewhat minimal in the prototype; further analysis is likely to yield improvements. Evaluation of the scoring mechanisms (understanding why appropriate sentences received lower scores than higher ranked sentences and understanding the contribution of the individual mechanisms) will also likely lead to improvements.

Since the prototype did not include a general search engine, the best interface with such systems is unknown. In addition, there are many applications that attempt to answer questions from specialized databases (such as FAQ databases, automatic message responders, and help files). There are also many specialized textual databases (historical records or genealogical databases). It seems that the prototype can work immediately with more or less static text databases, but in all these instances should also be able to take advantage of search functionality already included in such systems.

Some caveats are necessary in considering the results of the CL Research prototype and the possible improvements. Many of the questions in the TREC-8 Q&A track can possibly be better answered by simple lookup in dictionaries (including those that contain small amounts of encyclopedic information). Also, it appeared as if the phraseology of the questions frequently was very close to the answers in the text. The extent to which these considerations affect results needs to be determined.

7. Summary

The CL Research prototype system was reasonably successful in answering questions by selecting sentences from the documents in which the answers occur. The system generally indicates the viability of using relational triples (i.e., structural information in a sentence, consisting of discourse entities, semantic relations, and the governing words to which the entities are bound in the sentence) for question-answering. Post-hoc analysis of the results suggests several further improvements and the potential for investigating other avenues that make use of semantic networks and computational lexicology.

References

Fellbaum, C. (1998). *WordNet: An electronic lexical database*. Cambridge, Massachusetts: MIT Press.

Green, S. J. (1997). Automatically generating hypertext by computing semantic similarity [Diss], Toronto, Canada: University of Toronto.

Litkowski, K. C., & Harris, M. D. (1997). *Category development using complete semantic networks*. Technical Report, vol. 97-01. Gaithersburg, MD: CL Research.

Mani, I., & Bloedorn, E. (1999). Summarizing similarities and differences among related documents. *Information Retrieval, 1*(1), 35-68.

Richardson, S. D. (1997). Determining similarity and inferring relations in a lexical knowledge base [Diss], New York, NY: The City University of New York.