Question-Answering Using Semantic Relation Triples

Ken Litkowski
CL Research
ken@clres.com
http://www.clres.com
http://www.clres.com/trec8/index.html

CL Research Prototype System

- Sentence splitter that separated the source documents into individual sentences
- Parser which parsed each sentence into a parse tree containing the constituents of the sentence
- Parse tree analyzer to identify important elements and create semantic relation triples stored in a database
- Question-answering program
  - parsed the question into semantic relation triples, except with an unbound variable
  - matched the question database records with the document database to answer the question

Sentence Identification in Documents

- Combined parser sentence splitter with SGML markup to handle TREC documents (with need to reconstruct sentences)
- Extracted document identifier and document date and added question number and sentence number to provide unique identifier when questions were answered to extract the appropriate sentence from the document
- Only the top 10 documents (as ranked by the NIST search engine) were parsed
- Processed 1,977 documents from which 63,118 sentences were identified and presented to the parser
- An average of 31.9 sentences per document or 315.5 sentences in attempting to answer each question

Sentence Parsing

- **ATN-style grammar with 350 productions:**
  - start state, condition for transition, end state (with action functions to produce parse tree nodes and annotations)
  - adds parsing goals dynamically based on subcategorization patterns for lexical entries
  - parsing dictionary based on Oxford Advanced Learner's Dictionary (4th ed.)
- **Parsing dictionary is easily extensible**
- **Parses 400 definitions or 100 sentences per minute on 266 Mhz Pentium II with 64 MB RAM**
- **C source code available upon request (GNU GPL)**
  - compiles under Visual C++ 6.0, Borland C++ Builder3, Linux, BSD Unix, Sun4
  - 120 pages of documentation
- **Used by CL Research in Senseval (All-words category)**
  - 68% precision, 67% recall at coarse-grained level (highest)
  - Best overall improvement over baseline among all systems

Semantic Relation Triples

Full document parsing to identify discourse entities and their semantic roles in each sentence

- **Discourse entities**
  - Noun constituents, numbers, adjective sequences, possessives, leading noun sequences, ordinals, time phrases, predicative adjective phrases
- **Semantic relations**
  - Characterizes entities' semantic roles in sentence
  - Agent, theme, location, manner, modifier, purpose, and time (SUBJ, OBJ, prepositions heading prepositional phrases)
- **Governing words**
  - The word in the sentence that the discourse entity stood in relation to
  - For "SUBJ," "OBJ," and "TIME," the main verb of the sentence
  - For prepositions, the noun or verb that the prepositional phrase modified
  - For the adjectives and numbers, the noun that was modified

Document and Question Database Development

- Analyzed the parse tree of each sentence and question to
  - extract numbers, adjective sequences, possessives, leading noun sequences, ordinals, time phrases, predicative adjective phrases, conjuncts, and noun constituents as discourse entities
  - capture the semantic roles of the entities, as generally understood in linguistics, including agent, theme, location, manner, modifier, purpose, and time
  - identify the governing word for the discourse entity (the main verb, attachment point for prepositions, or noun modified
- 467,889 semantic relation triples were created in parsing the 63,118 sentences of 1,977 documents, an average of 7.4 triples per sentence
- 891 triples (for 196 questions), an average of 4.5 triples per question

Question Answering Routines

- **Matching the database records for an individual question against the database of documents for that question**
- **Coarse filtering of the records in the database to select potential sentences**
- **More refined filtering of the sentences according to the type of question**
- **Scoring the remaining sentences based on matches between the question and sentence database records**

Coarse Filtering

- **Question and sentence discourse entities were generally reduced to their root form**
- **All words were reduced to lowercase**
- **Allowed partial matching of words in multiword discourse entities from question database with discourse entities in document database**
- **Join between question and document databases produced an initial set of unique (document number, sentence number) pairs that were passed to refined filtering**

Refinement of Viable Sentences (1)

(Based on question-type, indicated by an unbound variable)

- **Time questions** (“when”, “what was the year” or “what was the date”): presence of TIME semantic relation (automatically assigned by parser, such as “last Thursday” or “in 1972"), with discourse entity containing an integer or having a word marked in the parser's dictionary as representing a time period, measurement time, month, or weekday
- **Location questions** (”where”): presence of “in” prepositional phrases
- **Who questions** (“who” or “whose”): searching for a record that had the same governing word (usually the main verb) as that of the unbound variable

Refinement of Viable Sentences (2)

(Based on question-type, indicated by an unbound variable)

- **What questions** (”what” or “which,” used alone or as question determiners, and unclassified questions, such as “why” or “name the”): searching for a record that had the same governing word (usually the main verb) as that of the unbound variable
- **Size questions** (“how” followed by an adjective): record that has a NUM semantic relation, examining governing word for measure, a unit, or a measurement size, with discourse entity and governing word as answer
- **Number questions** (“how many”): record that has a NUM semantic relation with the discourse entity in the question as the governing word

Sentence Scoring
(Each sentence begins with 1000)

- 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 discourse entity is a proper or complete substring of the discourse entity in the question record, 5 points added with semantic relation or governing word match; five points deducted if the match is not complete.
- If question discourse entity is a phrase, each word of the MWU is examined against the discourse entity in the sentence record, 5 points added for single word match, 20 for match on head noun, adding 5 points for semantic relation and governing word match
- Sentences that match on structural similarity will tend to separate out relative to other sentences in the documents

CL Research Prototype Results

- Official score for the CL Research 250-byte sentence submission was 0.281 (i.e., correct answer as 4th choice), compared to average of 0.332 among all submissions
- Only system that submitted full sentences
- Correct answers for 83 of the 198 questions
- Better than the median for 40 questions, equal for 109 questions, and less for 49 questions
- Performed better than the average score of 0.332 for 56 questions, 39 of which were “easier” questions and 17 of which were “harder” questions

Post-Hoc Analysis (1)

- General approach of matching relational structures between the questions and the documents is viable
  - selected 937 sentences and at least 83 correct sentences out of 63,177 sentences in the source documents
- Official results significantly understate the viability of the general approach in the prototype
  - only the top 10 documents were used (time constraints) in an attempt to answer the questions, when frequently an answer did not appear in any of these documents
  - 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) would result in a higher score
- More accurate overall score is approximately 0.482 (based on post-hoc analysis of 25 percent of the questions where no correct answer was provided)

Post-Hoc Analysis (2)
(Estimated Improvements)

- **Cutting off sentences (0.015)**: limitation to 250-byte strings cut off the portion that would have recognized by NIST evaluators as correct in first position
- **Inclusion of document containing answer (0.153)**: for one-third of wrongly-answered questions, the answer was not in the top 10 documents included in the database for the question; correct answers were identified in two-thirds of the cases, with an average inverse rank of 0.320
- **Relative time resolution (0.033)**: One-fifth of the questions answered incorrectly required resolution of relative time phrases ("last Thursday," "today," "two years ago"); available mechanisms not implemented had average inverse rank for the sentences provided in the prototype results is 0.292

Post-Hoc Analysis (3)

(Analysis of Other Wrongly-Answered Questions)

- For 6 percent, a sentence containing the correct answer was generated and was tied with an answer that was submitted
- For another 20 percent, the appropriate sentences were identified, but ranked lower than those submitted
- For another 17 percent, the answer required coreference resolution from one of the sentences submitted
- About 10 percent involved incomplete creation of document database entries due to bugs in the parser or in the mechanisms for extracting database records (these difficulties resulted in no sentences being submitted for 6 questions)

Conclusion and Future Work

- **Viability of using relational triples (i.e., structural information) demonstrated**
- **Dealing with problems identified in post-hoc analysis**
- **Incorporating (1) coreference resolution techniques and (2) named entity techniques**
- **Improvement of extraction techniques, particularly enhancing and generalizing semantic relations**
- **Discourse analysis techniques (such as text summarization research) for tying together records in the document databases**
- **Techniques from lexical cohesion and lexical chain research**
- **Improving mechanisms implemented in prototype (coarse filtering, analysis of question types, scoring routines)**