Experimental Study on the Impact of Concept Annotations on Semantic Search Performance

Naga Varun Dasari, IIT Bombay, India

ADVISORS: Arash Termehchy, Prof. Marianne Winslett

Background
- Research showed that when data are annotated by some semantic types, we get more relevant query results.
- But we don’t know how many and which semantic types have to be annotated for good query results.
- Also, we don’t know how relevant the query results would be if there were some errors in recognizing the concept annotations.

Goals
- The primary goal is to find which semantic types should be considered and annotated in the collection.
- The other goal is to study experimentally how the errors in concept annotations affect the precision and recall of the query results.

Fundamental Questions/Challenges
- If we index for each subset of concepts and query them to find which one gives the best results, it takes months to complete the process.
- We have to simulate the errors done by automatic recognizers by employing some error models without actually using automatic recognizers.

Experimental Setup
- Java is the programming language, with the Lucene library used for indexing and xerces for parsing.
- The corpus we worked on is an October 8, 2008 dump of Wikipedia collection and contains semantic annotations.

Plan (Goal 1)
- Select some concepts and find their probabilities of occurrence in the corpus.
- Find all subsets of concepts with a sum of probabilities less than a fixed value, say ‘d’.
- For each subset, manipulate the queries to give the same results they give when data are indexed with exactly this subset of concepts and perform queries.
- Find the subset that gives maximum relevant results and compare with the solution given by the existing greedy and brute-force algorithms.

Plan (Goal 2)
- Calculate the confidence and confusability lists of all concepts.
- For some combinations of precision and recall values, calculate the number of false positives (fp) and false negatives (fn) to be introduced in the corpus.
- Delete the bottom ‘fn’ entries in the confidence list from the corpus and add the top ‘fp’ entries in the confidence list into the corpus.
- Perform the queries and calculate the precision and recall of the results. Repeat the process for different initial precision and recall values.

Results
- Because of time limitations, we have been able to produce results only for the first goal. Below is a graph showing MAP vs. cost for the optimal solution and greedy and brute-force algorithms.

Future Work
- Complete goal 2 and produce results.
- Implement other error models and find which one best describes the automatic recognizers.
- Find the characteristics of the recognizers that affect the results.