The GURU System in TREC-5

Contact

. Yael Ravin
. Information Retrieval and Analysis
. T.J. Watson Research Center, H1-M09
. POB 704, Yorktown Heights, NY 10598
. yael@watson.ibm.com (914) 784-7847

System Description

The GURU information retrieval system is an experimental system developed at the T.J. Watson Research Center of IBM to support research in three main areas:

- Storage, indexing, and search of very large collections (hundreds of gigabytes) of full text documents. This includes research into distributed client/server environments and parallel processing.

- Mechanisms for querying the stored documents using free text and for ranking query results.

- Text analysis and information extraction tools for navigation, query preparation and query reformulation. This includes proper name identification, domain specific phrases and abbreviations, acronyms and their full forms (Ravin & Wacholder 1996).

GURU is a client/server system. The server includes an indexer, a search engine, a probabilistic ranking module, a query analysis module and a lexical server. Indexing is fairly standard. Text files are separated into documents, and each document is tokenized into individual words. All words, including stopwords, are indexed with their full position - paragraph, sentence, and word number. We are currently working on indexing proper names as well by integrating a module that identifies occurrences of different variants of the same name in the text and assigns them one canonical form.

At query time, one or more text collections to be searched is specified by the user. If multiple collections are specified, results from the different collections are merged before they are returned to the user. Our probabilistic ranking uses a unique feature called "Lexical Affinity" (LA). LA between two terms is a correlation measure of their common occurrences in text. as defined by Maareck (1991). The occurrences of correlated pairs of words in a document are ranked higher than the occurrences of the individual words over greater distances.

The analyzed query is expressed as an "F" statement. "F" is a formal language which we have developed for specifying different search operations and expansions of the query terms. For example: label1 { morph (word1) word2 } determines that the query consists of one query term (corresponding to one label). The curly brackets specify that word1 and word2 are variants of it. Occurrences of either variant will be treated by the search engine as occurrences of the query term. Variants are added manually by the user. The "morph" operator instructs the search engine to expand each word within its scope to all of its morphological forms. Morphological expansion is performed automatically by the engine on all query terms, but it can be over-ridden by the user.

The probabilistic ranking algorithm used by the search engines is based on work reported in Maarek and Smadja (1989) and Maarek (1991).

Participation in TREC-5

In TREC-5 we participated in the adhoc querying task. Our main goal is to optimize the probabilistic ranking algorithm used by GURU. Thus, we submitted four runs:

- Automatic query formulation with one ranking formula (labelled 'd');
- Automatic query formulation with another ranking formula (labelled 'e');
- Mamual query formulation with formula 'd';
- Mamual query formulation with formula 'e';

Due to various delays, we were unable to perform any training or fine-tuning for TREC. In addition, the person who formulated the queries for GURU was working remotely. As he was not familiar with the system and did not have sufficient time to communicate with us, the queries submitted in August were not properly formulated for the 'd' runs. We are currently re-running with new queries and will report in the final paper on any differences.

The GURU version with which we participated in TREC-5 is very simple: it uses a stopword list of about 250 words and applies automatic rule-based morphological expansion on the query terms. In TREC-5 we did not use proper names, phrases, or any other special data structures or knowledge bases. The system supports cross-collection searching -- the query is evaluated against each collection individually but the statistics are manipulated, to reflect the global statistics of all the collections searched.

References

Maarek, Y. S. "Software library construction from an IR perspective," in *SIGIR Forum*, Fall 1991, 25:2, 8-18.

Maarek Y. and F.A. Smadja "Full text indexing based on lexical relations. An application: software libraries." in N.J. Belkin and C.J. van Rijsbergen, eds, *Proceedings of SIGIR'89*, 198-206, Cambridge, MA, June 1989. ACM Press.

Ravin Y. and N. Wacholder, "Extracting Names from Natural-Language Text", IBM Research Report 20338, 1996.