Automatic Query Feedback using Related Words

Stefan M Rüger
Department of Computing
Imperial College of Science, Technology and Medicine
180 Queen's Gate, London SW7 2BZ, England
s.rueger@doc.ic.ac.uk

Abstract. Our experiments for the ad hoc task of TREC 8 were centered around the question how to create an automatic query feedback from the documents returned by an initial query.

1 The Query Process

1.1 Preprocessing of the Documents

In our retrieval experiments with $N = 528,155$ articles, we folded all words to lowercase and indexed all words with a document frequency of no more than 30% and which were not one of 349 stop words. For each document i we compute a weighted *document length* of

$$l_i = \sqrt{\sum_j (t_{ij} \log(0.3N/d_j))^2}, \tag{1}$$

where t_{ij} is the term frequency of word j in document i and d_j is the document frequency of word j.

Additionally, we identified roughly 100 *potentially interesting words* per document which we stored along with the meta-data of the document at index time. For these words we only kept nouns and adjectives based on Brill's tagger (Brill 1994) with a medium document frequency: the noun had to appear in least three documents and in no more that 30% of all documents. This resulted in a vocabulary of around 280,000 potentially interesting words. In our system we store a set of around 100 potentially interesting words per document along with the meta-data of the document at index time. Note that a set H of documents returned by a query may still have a potentially-interesting-words vocabulary of 10,000s of different words.

1.2 Processing of the Topics

We were only looking at the title and the description field of the topics, not at the narrative field. Each word of the title is included in the list of query words and also each non-stop word of the description field. For the description field, we had enhanced the stop words by about 60 function words of typical queries such as "relevant," "information" etc. Also, we decided to weigh the contribution of the words stemming from the description field with a factor of 1/5. A typical query would look like "blood-alcohol fatalities blood-alcohol:0.20 level:0.20 automobile:0.20 accident:0.20 fatalities:0.20".

1.3 First-Stage Document Retrieval

For each word j in the query list, we get the list of documents containing this word; we associate a score s_{ij} to each word j of document i in this list:

$$s_{ij} = q_j \cdot t_{ij} \cdot (\log(0.3N/d_j))^5 \,, \tag{2}$$

where q_j is the query weight (here 1.0 for words from the title and 0.2 for words from the description). Words with a document frequency of more than $0.3N$ had not been indexed and are associated to the empty document list. The exponent 5 is slightly non-standard (one would expect a 2), but our experiments have shown a beneficial ranking behaviour wrt earlier TREC queries.

We get another list of documents by broadening the query word using Porter's stemming and the same weighing scheme as above (2), d_j being the document frequency of the stem j and t_{ij} being the term frequency of the words with the stem j; we do not consider stems with a document frequency of more than $0.3N$. We repeat this process for all n query terms and, hence, arrive at $2n$ document lists. Subsequently, we create the union of these lists adding the scores of a particular document wrt all the query words and counting the number $n_i \leq 2n$ for each document, how many of the query terms and stems have been matched. We compute the final score of a document as

$$s_i = \left(\frac{n_i}{n}\right)^5 \Big(\sum_{j \in \text{query}} s_{ij} + \sum_{j \in \text{stems}} s_{ij} \Big) / l_i \tag{3}$$

The left factor rewards in a nonlinear way the documents that contain most of the query terms. Note that the score is normalized wrt the document length (1). The resulting document list is ordered according to the scores.

1.4 Related Words

The top-ranked 100 documents form a subset $H \subset D$ of the whole document set D. We suggest ranking the importance of each potentially interesting word j in the potentially-interesting-word vocabulary of H with a weight

$$w_j = \frac{h_j}{d_j} \cdot h_j \log(|H|/h_j),$$

where h_j is the number of documents in H containing the word j, and d_j is the document frequency of j wrt D. The second factor prefers medium matched-document frequency h_j, while the first factor prefers words that specifically occur in the matched documents. The highest-ranked words are meant to be related to the query. Indeed, we have something like "hardware", "software", "IBM" etc as the top-ranked words when querying for "computer". In the following, we use the top-ranked 50 words (according to w_j) which have a weight of not less than 2.5% of the maximum weight $\max_j(w_j)$. These words are called *related words*.

1.5 Automated Query Feedback

Another ranked document list is created from a related-words query, however, without stemming and with a slightly different ranking than (3):

$$s_i = \Big(\sum_{j \in \text{related}} s_{ij} \Big) / l_i \tag{4}$$

The scores of the related-words document list are normalized, such that the top score is 0.4. The scores of the resulting list of the first-stage document retrieval (see Subsection 1.3) are normalized so that the respective top score is 1.0. Now these two lists are joined, adding scores if necessary. This final list is sorted according to the joint score, cut off at 1000 and returned as result of the query.

2 TREC Evaluation and Conclusions

The ingredients to our information retrieval approach were ranked keyword retrieval using initial query words from the title, most words from the description of a topic and up to 50 related words wrt the results of an first-stage query. The above weighting and scoring schemes were picked using queries and relevant assessments of previous TRECs, though no systematic study has been carried out to optimize these schemes due to Black of time. During our preliminary studies, it was observed that query feedback improves precision and recall. Our scoring and ranking schemes seem rather ad hoc. We are convinced that a more systematic study could have improved the results considerably. One might want to lay out a more generic ranking scheme than the one above and introduce a reasonable number of parameters which are adjusted using training data and test data from previous TREC conferences and taking care of the potential pitfall of overfitting. This could lead to a potentially stronger ranking function, albeit one that is less applicable to theoretical reasoning. In addition, one might want to identify different classes of queries and appropriate ranking functions for each class to optimize the retrieval further. This is left to further studies.

References

Brill, E. (1994). Some advances in rule-based part of speech tagging. In *AAAI*.

Acknowledgements: This work was partially supported by the EPSRC, UK.