

Query Reformulation with Collaborative Concept-Based Expansion

Stefan Klink

German Research Center of Artificial Intelligence (DFKI)
P.O. Box 2080, 67608 Kaiserslautern, Germany

Stefan.Klink@dfki.de

ABSTRACT

Most information retrieval systems on the Internet try to supply the ‘best’ documents to a certain query by ranking the complete hit list with various algorithms. But this is only appropriate if the hit list contain relevant documents which can be positioned on the top. In the case of a small or empty hit list no ranking can improve the result and the user must reformulate the query again and again until he gets a sufficient result. This is a tedious task especially if the query is imprecise or too specific.

We propose a method for improving the original query by an automatic reformulation method. Each phrase of the query corresponds to a concept where similar terms are stored. These terms are used to reformulate the original query and the user directly receives a hit list with more relevant documents without numerous searching circles.

Keywords

Automatic query expansion, query reformulation, concepts

1. INTRODUCTION

The Internet has become more and more a popular medium for the exchange of information among various people. One special group in particular that has become increasingly dependent upon the Internet is the scientific community. For scientists it is common to search the information he needs e.g. scientific reports, papers, etc. through the Internet. Another rapidly raising group of people are non-scientific searching the weather-forecast, some information about some specific cars etc. and business people searching for information about traveling, hotels or some other business topics.

The problem of especially the latter group is that the terminology used in defining queries is often different to the terminology used in the representing documents. Many intelligent retrieval approaches [5],[12],[14] have tried to bridge this terminological gap.

One idea involves the use of a relevance feedback environment where the system retrieves documents that may be relevant to a user’s query. The user judges the relevance of one or more of the retrieved documents and these judgments are fed back to the system to improve the initial search result. This cycle of relevance feedback can be iterated until the user is satisfied with the retrieved documents. In this case, we can say that the more feedback is given to the system the better is the search effectiveness of the system. This behavior is verified by [4]. He has shown that the recall-precision effectiveness is proportional to the log of the number of relevant feedback documents.

But in a traditional relevance feedback environment the user voted documents are appropriate to the *complete* query. That means that the complete query is adapted to the users needs. If another user has the same intention but uses a different terminology or just one word more or less in his query then the traditional feedback environment doesn’t recognize any similarities in these situations.

Another idea to solve the terminology problem is to use query concepts. The system called **Rule Based Information Retrieval by Computer (RUBIC)** [1], [5], [12] uses production rules to capture user query concepts. In RUBIC, a set of related production rules is represented as an AND/OR tree, called a rule base tree. RUBIC allows the definition of detailed queries starting at a conceptual level. The retrieval output is determined by fuzzy evaluation of the AND/OR tree. To find proper weight values, Kim and Raghavan developed a neural network (NN) model in which the weights for the rules can be adjusted by users’ relevance feedback. Their approach is different from the previous NN approaches for IR in two aspects [8]. They handle relations between concepts and Boolean expressions in which weighted terms are involved. Second, they do not use their own network model but an already proven model in terms of its performance.

But the crucial problem of a rule-based system still exists: the automatic production of proper rules and the learning of appropriate structures of rules, not just the weights.

2. QUERY EXPANSION

The crucial point in query expansion is the question: Which terms (phrases) should be included in the query formulation.? If the query formulation is to be expanded by additional terms there are two problems that are to be solved namely:

1. how are these terms selected and
2. how are the parameters estimated for these terms

For the selection task mainly three different strategies have been proposed:

- Dependent terms: Here terms that are dependent on the query terms are selected. For this purpose the similarity between all terms of the document collection has to be computed first [14].
- Feedback terms: From the documents that have been judged by the user the most significant terms (according to a measure that considers the distribution of a term within relevant and non-relevant documents) are added to the query formulation[17].
- Interactive selection: By means of one of the methods mentioned before a list of candidate terms is computed

and presented to the user who makes the final decision which terms are to be included in the query[6].

Due to nowadays results available so fare indicate that clear improvements are reported in [17] and [10] for the feedback terms method.

Many terms used in human communication are ambiguous or have several meanings [14]. But in most cases these ambiguities are resolved without any problem, or even without noticing the ambiguity. The way this is done by humans is still an open problem of psychological research, but it is almost certain, that the context in which a term occurs plays a central role.

The expansion of single terms of a query ignores the context in which a term is used. There is no way that influences coming from several terms of the query can accumulate in a term or that negative associations coming from a query term can lower the probability of another term to be selected for expansion. A very simple way to allow such influences is a linear model, that results in a superposition of the influences coming from all query terms.

3. USING CONCEPTS

A related aspect is the size of the query which should be expanded. If the query consists of only a few terms, then this superposition of influences is rather limited. Users of text retrieval systems including searchers on the Web consistently generate brief and often broad two or three term queries and have proven to be very reluctant to explore current tools to expand initial queries [12].

In our approach, we cut the complete query into several phrases (q_1, \dots, q_n), where each phrase consists of one or more words. Now, the system identifies each phrase q_i with a concept $C(q_i)$ and the original query Q is expanded with the phrases of each appropriate concept (see Figure 1).

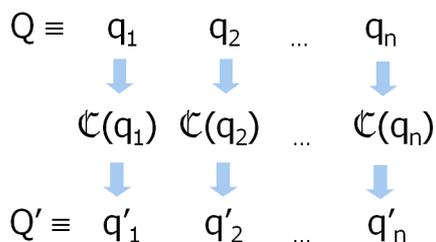


Figure 1. Expansion of the user query with concepts

The vital advantage of using concepts and not learning the complete query is that other users can profit from the learned concepts even if their query is not completely the same. A statistical evaluation of log files made by our group has shown that the probability that a searcher uses exactly the same query than a previous searcher is much lower then the probability that parts of the query (phrases or terms) occurs in other queries.

So, even if a web searcher never used the given search phrase, the probability that other searcher had used it is very high and then he can profit from a stored concept.

4. SEARCHING WITH CONCEPTS

At the first phase of the searching process, the user gives the system an initial query and the phrases are identified.

If for each phrase a concept is found, then each of them are expanded with the phrases of the appropriate concept and the user given query is reformulated as described before.

But if for one phrase no concept could be found, then it is the decision of the user, if the phrase should not be expanded or if he will use the system. In the latter case, he gets the hit list of the search machine which is appropriate to his original query (see Figure 2).

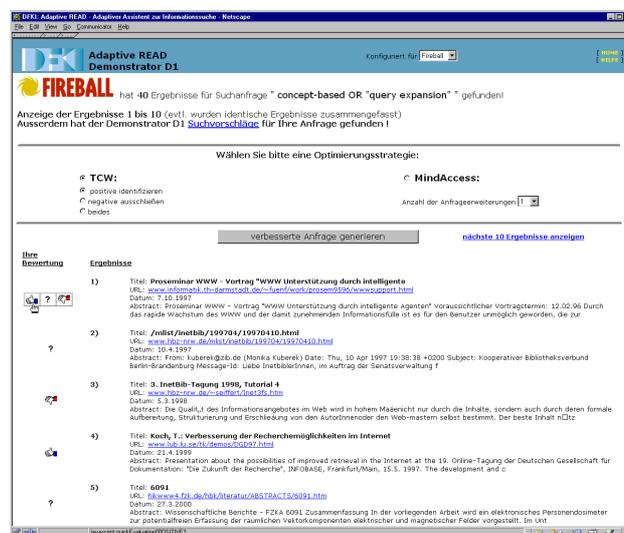


Figure 2. Graphical user front-end for giving feedback

The system uses the feed-back to learn the concepts and does not need the complete document corpus. For doing this, the searcher is able to vote the documents in the hit list with “thumb up”, “thumb down” or “questionable” to indicate a good or a bad document. The third state (“questionable”) is to say that this document is not relevant and should not be used by the system for learning the concept. The URLs are clickable for viewing the actual documents so that the user can make the judgements more accurately. Doing this, the selected document is shown in a separate window with extra voting buttons (“good”, “bad”, “questionable”). The voting is directly transmitted to the user front-end and the appropriate document gets the voting sign.

After voting all desired documents, the searcher clicks the button “verbesserte Anfrage generieren” (“generate a better query”) and the system loads all voted documents directly from the original URL. It uses not only the abstract or a document index but the complete documents to learn the new concepts for the non-found phrases. After storing these concepts in a concept database, the system is able to expand each phrase and the original query of the user is reformulated as described before.

The new query is presented to the searcher to confirm the reformulation. After that, the query is sent to the search machine and the searcher will get a new hit list with relevant documents.

5. CONCEPT DATABASE HIERARCHIES

By default, the concepts are stored in a global concept database which is accessible by each web searcher and each searcher uses these concepts. But in some cases, it could be advantageous that an individual user or a group of users have their own concepts. For example, if all members of a company have his own terminology for some things. Or if you think of languages each country has his own language and his own terminology. The user can select any desired (user, group, etc.) profile he want to use and if a concept is not found in the selected profile then the system searches for this concept in the profiles which are located higher in the hierarchy up to the global concept database (see Figure 3).

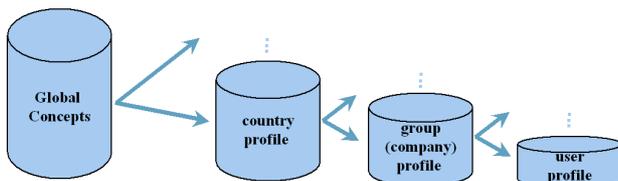


Figure 3. Hierarchy of concept databases

If the individual user does not agree with a given concept, he can learn his own one with the judged documents.

6. LEARNING CONCEPTS

As described above, each phrase of the original query is expanded with the appropriate concept and if no concept could be found, the user is ask to give some feed-back on relevant and non-relevant documents. With the help of these voted documents the new concept is created. This is done by adding the most informative words as features and the newly learned concept is stored in the concept database which the user has chosen.

The more interesting part is to improve concepts which are already stored in the concept database. In the case the user does not agree with a given concept, he can decide, if he wants to throw it away and the concept is created by the voted documents or the system can use the given concept as a starting point for a revision task.

But for a revision task, the system needs to determine to what degree it should believe in the user's estimates and to what extent they should be updated when voted documents are encountered. This decision is related to the amount of available documents. When the user just has voted a few documents, the system rely more on the given concept than on estimates formed by looking at only the available voted documents. As more voted documents become available, the system gradually increase the belief in probability estimates from the voted documents and gradually decrease the weight of the given concept.

A theoretically sound way of doing this is to express the uncertainty in a probability estimate with a conjugate probability distribution. While observing more voted documents, this probability distribution can be updated to reflect both the changed expected value of the probability, and the increased confidence in the accuracy of the probability estimate.

Conjugate priors are a technique from Bayesian statistics to update probabilities from data [7]. In our system, the equivalent sample size approach is used and by default the weight of the prior probability estimate is equivalent to 50 samples. For example, if the concept is based on 50 voted relevant pages and 40 of them contain the word within the concept, then the probability estimate is 0.8. After voting 25 additional relevant pages and 10 of them contain the word, then the value would be 50/75.

7. CONCLUSION AND PROSPECTS

We have described an approach for bridging the gap of different terminology within the user query and the searched documents. Each term of the query corresponds to a concept which is learned from the documents given by the feedback of the actual or of other users. The vital advantage in our approach is that each user can profit from the concepts learned by other users.

On our experiments we get promising results in regard of document precision and in regard of document recall (reducing the huge hit list, e.g. 11057 hits reduced to 275).

The next step will be to make some experiments using concepts within the hierarchy of a larger group of users and evaluating the effects on the recall and precision.

Another interesting aspect would be to make some research on information shifting when learning a concept. The more often a concept is revisited the more stable it will be. But a problem arises when the user shifts his interests. Then he must throw away his learned concept and must learn it from scratch. In this situation it would be useful to 'search' for similar concepts within the hierarchy of concept databases and present it to the user to support him.

One important area for future work is to carry out the same type of analysis on a larger scale. The first way is to enlarge the data set. We are planning to use the WT10g collection (TREC) to validate our experiments made so far. And the second way is to enlarge the set of users. This enables a better determination of the extent to which experts agree and provides a better target for evaluating algorithms; the most plausible way to get more experts is to do a distributed, web-based experiments. The home page of our system is coming soon...

8. REFERENCES

- [1] Aalbersberg I.J.: *Incremental relevance feedback*. In Proceedings of the Annual Int. ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 11 - 22, 1992
- [2] Allan J.: *Incremental relevance feedback for information filtering*. In Proceedings of the Annual Int. ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 270 - 278, 1996
- [3] Alsaffar A.H., Deogun J.S., Raghavan V.V., Sever H: *Concept-based retrieval with minimal term sets*. In Z.W. Ras and A. Skowon, editors, Foundation of Intelligent Systems: 11th Int. Symposium, ISMIS'99 proceedings, pp. 114-122, Springer, Warsaw, Poland, Jun 1999.
- [4] Buckley C., Salton G., Allen J.: *The effect of adding relevance information in a relevance feedback environment*. In Proceedings of the Annual Int. ACM SIGIR Conference on

- Research and Development in Information Retrieval, pp. 292 - 300, 1994
- [5] Croft W.B.: *Approaches to intelligent information retrieval*. Information Processing and Management, 1987, Vol.23, No.4, pp. 249-254
- [6] Harman D.: *Towards Interactive Query Expansion*. In: Chiaramella Y. (editor): 11th International Conference on Research and Development in Information Retrieval, pp. 321 - 331, Grenoble, France, 1988
- [7] Heckerman, D.: *A Tutorial on Learning with Bayesian Networks*. (Technical Report MSR-TR-95-06). Microsoft Corporation, 1995.
- [8] Iwayama M.: *Relevance Feedback with a Small Number of Relevance Judgements: Incremental Relevance Feedback vs. Document Clustering*. In Proceedings of the 23rd Annual Int. ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 10 - 16, Athens, Greece, July 2000
- [9] Kim M., Raghavan V.: *Adaptive concept-based Retrieval Using a Neural Network*, In Proceedings of ACM SIGIR Workshop on Mathematical/Formal Methods in Information Retrieval, Athens, Greece, July 2000
- [10] Kwok K.: *Query Modification and Expansion in a Network with Adaptive Architecture*. In Proceedings of the 14th Annual Int. ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 192 - 201, 1991
- [11] Lu F., Johnsten Th., Raghavan V.V., Traylor D.: *Enhancing Internet Search Engines to Achieve Concept-based Retrieval*, Proceeding of Inforum'99, Oakridge, USA
- [12] Maglano V., Beaulieu M., Robertson S.: *Evaluation of interfaces for IRS: modelling end-user search behaviour*. 20th Colloquium on Information Retrieval, Grenoble, 1988
- [13] McCune B.P., Tong R.M., Dean J.S., Shapiro D.G.: *RUBRIC: A System for Rule-Based Information Retrieval*, IEEE Transaction on Software Engineering, Vol. SE-11, No.9, September 1985
- [14] Pirkola A.: *Studies on Linguistic Problems and Methods in Text Retrieval: The Effects of Anaphor and Ellipsis Resolution in Proximity Searching, and Translation and query Structuring Methods in Cross-Language Retrieval*, PhD dissertation, Department of Information Studies, University of Tampere. Acta Universitatis Tampereensis 672. ISBN 951-44-4582-1; ISSN 1455-1616. June 1999
- [15] van Rijsbergen C.J., Harper D.H., et al.: *The Selection of Good Search Terms*. Information Processing and Management 17, pp. 77-91, 1981
- [16] Resnik P.: *Using information content to evaluate semantic similarity in a taxonomy*. In Proceedings of the 14th Int. Joint Conference on Artificial Intelligence, pp. 448-453, 1995
- [17] Salton G., Buckley C.: *Improving Retrieval Performance by Relevance Feedback*. Journal of the American Society for Information Science 41(4), pp. 288 - 297, 1990
- [18] Sanderson M., Croft B.: *Deriving concept hierarchies from text*. In Proceedings of the 22nd Annual Int. ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 206 - 213, Berkeley, CA, August 1999
- [19] Stucky D.: *Unterstützung der Anfrageformulierung bei Internet-Suchmaschinen durch User Relevance Feedback*, diploma thesis, German Research Center of Artificial Intelligence (DFKI), Kaiserslautern, November 2000