Collaborative Learning of Term-based Concepts for Automatic Query Expansion

Stefan Klink, Armin Hust, Markus Junker, and Andreas Dengel

German Research Center for Artificial Intelligence (DFKI, GmbH)
P.O. Box 2080, 67608 Kaiserslautern, Germany
{Stefan.Klink, Armin.Hust, Markus.Junker,
Andreas.Dengel}@dfki.de
http://www.dfki.de/~klink

Abstract. Information Retrieval Systems have been studied in Computer Science for decades. The traditional ad-hoc task is to find all documents relevant for an ad-hoc given query but the accuracy of ad-hoc document retrieval systems has plateaued in recent years. At DFKI, we are working on so-called collaborative information retrieval (CIR) systems which unintrusively learn from their users search processes.
In this paper, a new approach is presented called term-based concept learning (TCL) which learns conceptual description terms occurring in known queries. A new query is expanded term by term using the previously learned concepts.
Experiments have shown that TCL and the combination with pseudo relevance feedback result in notable improvements in the retrieval effectiveness if measured the recall/precision in comparison to the standard vector space model and to the pseudo relevance feedback. This approach can be used to improve the retrieval of documents in Digital Libraries, in Document Management Systems, in the WWW etc.

1 Introduction

With the explosive growth of information on the Internet and Digital Libraries, an acute problem has raised called information overload. Typical search engines index billions of pages across a variety of categories, and return results ranked by expected topical relevance. But only a small percentage of these pages may be of a specific interest. Nowadays, there is an acute need for search engine technology to help users exploit such an extremely valuable resource.

In weighted Information Retrieval (IR) the number of retrieved documents is related to the number of appropriate search terms. Retrieval with short queries is typical in Web search [6], but it is much harder as compared to retrieval with long queries. This is because shorter queries often provide less information for retrieval. Modern IR systems therefore integrate thesaurus browsers. They help to find additional search terms [13]. But the keywords used in short queries are not always good descriptors of contents. Nevertheless, most existing search engines still rely solely on the keywords contained in queries to search and rank relevant documents. This is one of the key reasons that affect the precision of the search engines. In many cases, the answer

documents are not relevant to the user information need, although they do contain the same keyword as the query.

Another problem which is typical for the Web and for Digital Libraries is that the terminology used in defining queries is often different to the terminology used in the representing documents. Even if some users have the same information need they rarely use the same terminology in their queries. Many intelligent retrieval approaches [2, 8, 12] have tried to bridge this terminological gap.

Research on automatic query expansion (or modification) was already under way before the 60's when initial requests were enlarged in the grounds of statistical evidence [14]. The idea was to obtain additional relevant documents through expanded queries based on the co-occurrence of the terms. However, this kind of automatic query expansion has not been very successful. The retrieval effectiveness of the expanded queries was often not greater than, or even less that the effectiveness of the original queries [10, 11, 16].

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 [1]. 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.

2 Query Expansion

The crucial point in query expansion is the question: Which terms (or 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 how are these terms selected and how are the parameters estimated for these terms.

Many terms used in human communication are ambiguous or have several meanings [12]. But in most cases these ambiguities are resolved automatically 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.

Most attempts at automatically expanding queries failed to improve the retrieval effectiveness and it was often concluded that automatic query expansion based on statistical data was unable to improve the retrieval effectiveness substantial [11].

But this could have several reasons. Term-based query expansion approaches are mostly using hand-made thesauri or just plain co-occurrence data. They do not use

learning technologies for the query terms. On the other hand, those who use learning technologies (Neural Networks, Support Vector Machines, etc.) are query-based. That means these systems learn concepts (or additional terms) for the complete query.

The vital advantage of using term-based concepts and not learning the complete query is that other users can profit from the learned concepts. A statistical evaluation of internet logging files 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 term, the probability that another searcher had used it is very high and then he can profit from the learned concept.

3 Traditional Document Retrieval

The task of traditional document retrieval is to retrieve documents which are relevant to a given query from a fixed set of documents. Documents as well as queries are represented in a common way using a set of index terms (called terms from now on). Terms are determined from words of the documents in the database, usually during pre-processing phases where some noise reduction procedures are incorporated, e.g. stemming and stop-word elimination.

In the following, a term is represented by t_i ($1 \le i \le M$) and a document by $\boldsymbol{d}_j$ ($1 \le j \le N$), respectively, where M is the number of different terms and N is the number of documents in the data base.

3.1 Vector Space Model

One of the most popular and indeed the simplest retrieval model is the vector space model (VSM) [2]. In the VSM, each document $\boldsymbol{d}_j$ is represented as a M dimensional vector

$$\boldsymbol{d}_j = (w_{1j}, \ldots, w_{Mj})^T, \; 1 \le j \le N \qquad (\mathbf{1})$$

where T indicates the transpose, w_{ij} is a weight of a term t_i in a document $\boldsymbol{d}_j$. A query is likewise represented as

$$\boldsymbol{q}_k = (w_{1q}, \ldots, w_{Mq})^T, \quad 1 \le k \le L \qquad (\mathbf{2})$$

where w_{iq} is a weight of a term t_i in a query $\boldsymbol{q}_k$ and L is the number of queries contained in the document collection (e.g. in the web-log file).

The weights above can be processed in various ways. In our approach, we use the standard normalized tf · idf weighting scheme [14] defined as follows:

$$w_{ij} = tf_{ij} * idf_i \qquad (\mathbf{3})$$

where tf_{ij} is the weight calculated using the term frequency f_{ij} and idf_i is the weight calculated using the inverse of the document frequency.

The result of the retrieval is represented as a list of documents ranked according to their similarity to the given query. The similarity $sim(\boldsymbol{d}_j, \boldsymbol{q}_k)$ between a document $\boldsymbol{d}_j$ and a query $\boldsymbol{q}_k$ is measured by the standard cosine of the angle between these M dimensional vectors $\boldsymbol{d}_j$ and $\boldsymbol{q}_k$:

$$sim(\boldsymbol{d}_j, \boldsymbol{q}_k) = \frac{\boldsymbol{d}_j^T \boldsymbol{q}_k}{\|\boldsymbol{d}_j\| \ \|\boldsymbol{q}_k\|} \tag{4}$$

where $\|\cdot\|$ is the Euclidean norm of a vector. In the case that the vectors are already normalized (e.g. have a unit length) the similarity is just the dot product between the two vectors.

The VSM is one of the methods we applied to compare with our own methods.

3.2 Pseudo-Relevance-Feedback

Everybody would agree that documents my be relevant to a query even if they do not share any word with the query. Unfortunately, the standard VSM will always return zero similarity in this case. So-called query expansion techniques try to overcome this problem by expanding the user given query $\boldsymbol{q}$ to a new enriched query $\boldsymbol{q}$' which hen used in the standard VSM. A very well-known technique is the pseudo relevance feedback (PRF) [8]. PRF enriches the original query $\boldsymbol{q}$ by the terms of the top-ranked documents with respect to $\boldsymbol{q}$. We are using a variation of PRF described in [7]:

Let $\mathbb{E}$ be a set of document vectors for expansion given by

$$\mathbb{E} = \left\{ \boldsymbol{d}_j^+ \;\middle|\; \frac{\text{sim}(\boldsymbol{d}_j^+, \boldsymbol{q})}{\max_i \text{sim}(\boldsymbol{d}_i, \boldsymbol{q})} \geq \theta \right\} \tag{5}$$

where $\boldsymbol{q}$ is the users query vector and θ is a similarity threshold. The sum $\boldsymbol{d}_s$ of the document vectors in $\mathbb{E}$:

$$\boldsymbol{d}_s = \sum_{\boldsymbol{d}_j^+ \in \mathbb{E}} \boldsymbol{d}_j^+ \tag{6}$$

can be considered as enriched information about the original query. Within the VSM, this sum of documents is again a vector which contains the weights of all terms of the summed documents. Hence, all terms of the documents in $\mathbb{E}$ are used to expand the users query. The expanded query vector $\boldsymbol{q}'$ is obtained by

$$\boldsymbol{q}' = \frac{\boldsymbol{q}}{\|\boldsymbol{q}\|} + \alpha \frac{\boldsymbol{d}_s}{\|\boldsymbol{d}_s\|} \tag{7}$$

where α is a parameter for controlling the weight of the newly incorporated terms. Finally, the documents are ranked again according to the similarity $sim(\boldsymbol{d}_j, \boldsymbol{q}')$ to the expanded query.

4 Term-based Concept Learning (TCL)

A problem of the standard VSM is that a query is often too short to rank documents appropriately. To cope with this problem, the approach is to enrich the original query with terms occurring in the documents of the collection.

Our method uses feedback information and information globally available from previous queries. Feedback information in our environment is available within the ground truth data provided by the test document collections. The ground truth provides relevance information, i.e. for each query a list of relevant documents exists.

Relevance information for each query is represented by a N dimensional vector:

$$\boldsymbol{r}_k = (r_{1k}, \ldots, r_{Nk})^T \quad , \quad r_{jk} = \begin{cases} 1, \text{ if docment } \boldsymbol{d}_j \text{ is relevant to query } k \\ 0, \text{ if docment } \boldsymbol{d}_j \text{ is not relevant to query } \boldsymbol{q}_k \end{cases} \tag{8}$$

In contrast to traditional pseudo relevance feedback methods, where the top j ranked documents are assumed to be relevant and then their terms are incorporated into the expanded query, we use a different technique to compute the relevant documents. Our method is divided into two phases:

The **learning phase** for each term works as follows:

- Select the old queries in which the specific query term occurs
- From these selected old queries get the sets of relevant documents from the ground truth data
- From each set of relevant documents compute a new document vector and use these document vectors to build the term concept.

The **expansion phase** for each term is easy:

- Select the appropriate concept of the current term
- Use a weighting scheme to enrich the new query with the concept

For the formal description of the learning phase we need the following definitions:

- $\boldsymbol{D} = \{\boldsymbol{d}_1, \ldots, \boldsymbol{d}_N\}$: the set of all documents
- $\boldsymbol{Q} = \{\boldsymbol{q}_1, \ldots, \boldsymbol{q}_L\}$: the set of all known queries with
- $\boldsymbol{q}_k = (w_{1k}, \ldots, w_{ik}, \ldots, w_{Mk})^T$ represented within the vector space model. For each term of the query the appropriate weight w_{ik} is between 0 an 1.
- $\boldsymbol{R}^+(\boldsymbol{q}_k) = \{\boldsymbol{d}_j \in \boldsymbol{D} \mid r_{ij} = 1\}$: the set of all documents relevant to the query $\boldsymbol{q}_k$

Now, the first step of the learning phase collects all queries having the i-th term in common:

$$\boldsymbol{Q}_i = \{ \boldsymbol{q}_k \in \boldsymbol{Q} \mid w_{ik} \neq 0 \} \tag{9}$$

If the i-th term doesn't occur in any query $\boldsymbol{q}_k$ then $\boldsymbol{Q}_i$ is empty.
The second step collects all documents which are relevant to these collected queries:

$$\boldsymbol{D}_{ik} = \{\boldsymbol{d}_j \mid \boldsymbol{d}_j \in \boldsymbol{R}^+(\boldsymbol{q}_k) \wedge \boldsymbol{q}_k \in \boldsymbol{Q}_i \} \tag{10}$$

In the last step of the learning phase the concept of each *i*-th term is build as the sum of all documents (i.e. vectors of term weights) which are relevant to the known queries which have the term in common:

$$\boldsymbol{C}_i = \sum_{\boldsymbol{d}_j \in \boldsymbol{D}_{ik}} \boldsymbol{d}_j \tag{11}$$

As queries and documents, a concept is represented by a vector of term weights. If no query $\boldsymbol{q}_k$ contains term *i*, the corresponding concept $\boldsymbol{C}_i$ is represented as $(0,\ldots,0)^T$.

Now, where the term-based concepts are learned, the user query $\boldsymbol{q}$ can be expanded term by term. The expanded query vector $\boldsymbol{q'}$ is obtained by

$$\boldsymbol{q'} = \boldsymbol{q} + \sum_{i=1}^{M} \omega_i \boldsymbol{C}_i \tag{12}$$

where ω_i are parameters for weighting the concepts. In the experiments described below ω_i is set to 1. Before applying the expanded query, it is normalized by

$$\boldsymbol{q''} = \frac{\boldsymbol{q'}}{\|\boldsymbol{q'}\|} \tag{13}$$

For this approach, the complete documents (all term weights w_{ij} of the relevant documents) are summed up and added to the query. Although, in some papers it is reported that using just the top ranked terms is sufficient or sometimes better. But experiments with this approach on the collections have shown that the more words are used to learn the concepts the better the results are. So, the decision was made to use always the complete documents and not only some (top ranked) terms.

If no ground truth of relevant documents is available, relevant feedback techniques can be used and the concepts are learned by adding terms from the retrieved relevant documents.

5 Combination of PRF with TCL

Additionally to the approach described above, we made some experiments with a linear combination of our approach with the pseudo relevance feedback. For each query we applied PRF method in parallel to our method and the new query is build by (cmp. (**7**) and (**12**)):

$$\boldsymbol{q'} = \boldsymbol{q} + \beta \boldsymbol{d}_s + \sum_{i=1}^{M} \omega_i \boldsymbol{C}_i \tag{14}$$

Before applying the expanded query, it is normalized by equation (**13**).

6 Experiments and Results

6.1 Test Collections

For our comparison we used four standard test collections: CACM (collection of titles and abstracts from the journal 'Communications of the ACM'), CR (congressional report), FR88 (federal register), NPL (also known as the VASWANI) and ZF3 ('Computer Select' of Ziff-Davis Publishing Co.). These collections are contained in the TREC disks [4]. All collections are provided with queries and their ground truth (for each query a list of relevant documents). For these collections, terms used for document representation were obtained by stemming and eliminating stop words.

Table 1. Statistics about collections after stemming and stop words elimination

	CACM	CR	FR88	NPL	ZF3
# documents	3204	27922	19860	11429	161021
# queries	52	34	199	93	50
# different terms	3029	45717	43765	4415	67108
avg doc length [terms]	25.8	672.8	869.1	21.8	155.6
avg query length [terms]	10.9	3.1	3.5	6.6	7.7

In addition to the number of documents, a significant difference is the length of the documents: CACM and NPL consists of abstracts, while CR, FR88, and ZF3 contain (much) longer documents.

Queries in the TREC collections are mostly provided in a structured format with several fields. In this paper, the "title" (the shortest representation) is used for the CR and NPL collection whereas the "desc" (description; medium length) is used for the CACM, FR88, and ZF3 collection. The short queries in FR88 are obtained by stop word elimination

6.2 Evaluation

The following paragraphs describe some basic evaluation methods used in this paper. For further information and a more detailed description see Kise et al [7].

6.2.1 Average Precision

A common way to evaluate the performance of retrieval methods is to compute the (interpolated) precision at some recall levels. This results in a number of recall/precision points which are displayed in recall-precision graphs [2]. However, it is sometimes convenient for us to have a single value that summarizes the performance. The average precision (non-interpolated) over all relevant documents [2, 3] is a measure resulting in a single value. The definition is as follows:

As described in section 3, the result of retrieval is represented as the ranked list of documents. Let $r(i)$ be the rank of the i-th relevant document counted from the top of the list. The precision for this document is calculated by $i/r(i)$. The precision values

for all documents relevant to a query are averaged to obtain a single value for the query. The average precision over all relevant documents is then obtained by averaging the respective values over all queries.

6.2.2 Statistical Test

The next step for the evaluation is to compare the values of the average precision obtained by different methods [7]. An important question here is whether the difference in the average precision is really meaningful or just by chance. In order to make such a distinction, it is necessary to apply a statistical test.

Several statistical tests have been applied to the task of information retrieval [5,17]. In this paper, we utilize the test called "macro t-test" [17] (called paired t-test in [5]). The following is a summary of the test described in [7]:

Let a_i and b_i be the scores (e.g., the average precision) of retrieval methods A and B for a query i and define $d_i = a_i - b_i$. The test can be applied under the assumptions that the model is additive, i.e., $d_i = \mu + \varepsilon_i$ where μ is the population mean and ε_i is an error, and that the errors are normally distributed. The null hypothesis here is $\mu = 0$ (A performs equivalently to B in terms of the average precision), and the alternative hypothesis is $\mu > 0$ (A performs better than B).
It is known that the Student's t-statistic

$$t = \frac{\bar{d}}{\sqrt{s^2 / n}} \quad with \quad \bar{d} = \frac{1}{n} \sum_{i=1}^{n} d_i \quad and \quad s^2 = \frac{1}{n-1} \sum_{i=1}^{n} (d_i - \bar{d})^2 \qquad (\mathbf{15})$$

follows the t-distribution with the degree of freedom of $n - 1$, where n is the number of samples (queries), $\bar{d}$ and s^2 are the sample mean and the variance.
By looking up the value of t in the t-distribution, we can obtain the P-value, i.e., the probability of observing the sample results d_i ($1 \le i \le n$) under the assumption that the null hypothesis is true. The P-value is compared to a predetermined significance level σ in order to decide whether the null hypothesis should be rejected or not. As significance levels, we utilize 0.05 and 0.01.

6.3 Results and Comparison to the Standard

6.3.1 Recall and Precision

The results of the pseudo relevance feedback are depending on two parameters α (weight) and θ (similarity threshold). To get the best results, we were varying α from 0 to 5.0 with step 0.1 and θ from 0.0 to 1.0 with step 0.05. For the combined approach we calculated the best β by varying from 0 to 1.0 with step 0.1.

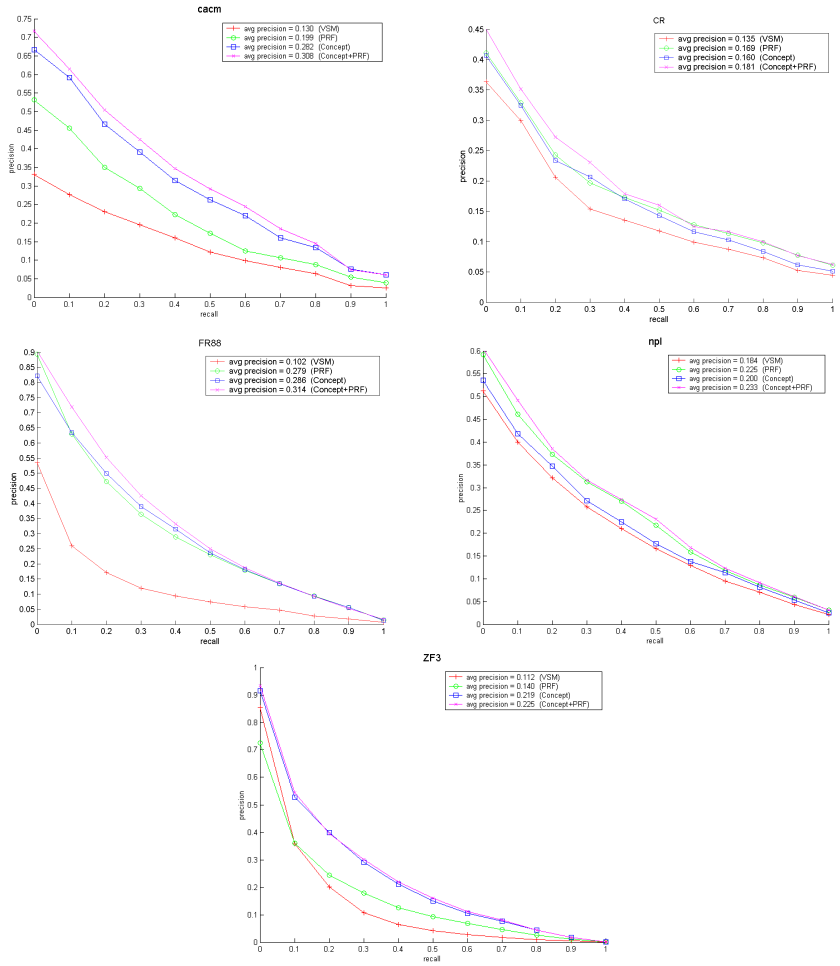
For each collection, the best individual α, θ, and β are calculated and used for the comparison. Table 2 shows the best values for each collection:

Table 2. Best values for pseudo relevance feedback and the combined method parameters

	CACM	CR	FR88	NPL	ZF3
α / θ (PRF)	1.70 / 0.35	0.6 / 0.75	0.60 / 0.00	2.00 / 0.45	0.80 / 0.50
β (comb.)	0.4	0.6	0.4	0.9	0.2

The results of our concept-based expansion are also depending on weights. But due to time restrictions, we had not enough time to vary the weights in a range for each collection. We just used the default value: $\omega_i = 1$, which means that the original term and the learned concept are weighted equal.

Figure 1 shows the recall/precision results of the original query with the standard vector space model (VSM), the pseudo relevance feedback (PRF), the expanded query using TCL (Concepts) and the combination (Concepts+PRF):



F ig. 1. recall / precision of CACM, CR, FR88, NPL, and ZF3

The recall/precision graphs in figure 1 indicate that the automatic query expansion method based on learned concepts yields a considerable improvement in the retrieval effectiveness in mostly all collections over all recall points compared to the standard vector space model and to the pseudo-relevance-feedback method (except with the NPL collection). There is no indication that the improvement is depending on the size of the collection, the number of documents nor on the number or size of the queries. On the small ones the method performs good on CACM but only somewhat better than the VSM on the NPL and on the FR it performs better than on the CR collection. On a closer look at the figures the impression could arise that our approach performs better on longer queries. But experiments with the CR collection have shown that 'title' queries result a better precision than 'description' or 'narrative' queries. This behavior is in contrast to the first impression of the figures.

6.3.2 Statistical tests

To be sure that these exciting results are really meaningful and not just by chance, it is necessary to apply a statistical test. As described above, we used the "macro t-test". The results of this test for all pairs of methods are shown in table 3. The meaning of the symbols such as "≫", ">" and "~" is summarized at the bottom of the table. For example, the symbol "<" was obtained in the case of the concept method compared to VSM for the NPL collection. This indicates that (at significance $\sigma = 0.05$) the null hypothesis "concept method performs equivalently to the VSM" is rejected and the alternative hypothesis "concept method performs better than the VSM" is accepted. (At $\sigma = 0.01$, however, the null hypothesis cannot be rejected.)

Roughly speaking, "A ≫(≪) B", "A > (<) B" and "A~B" indicate that "A is almost guaranteed to be better (worse) than B", "A is likely to be better (worse) than B" and "A is equivalent to B", respectively.

Table 3. Results of the macro t-test.

methods (A vs. B)	CACM	CR	FR88	NPL	ZF3
PRF vs. VSM	≫	~	≫	≫	≫
Concepts vs. VSM	≫	≫	≫	>	≫
Concepts vs. PRF	≫	>	~	<	≫
Concepts+PRF vs. VSM	≫	≫	≫	≫	≫
Concepts+PRF vs. PRF	≫	>	≫	~	≫
Concepts+PRF vs. concepts	>	~	≫	≫	~

≫, ≪ : P-value ≤ 0.01
>, < : 0.01 ≤ P-value ≤ 0.05
~ : 0.05 ≤ P-value

The macro t-tests prove our results. Our new method for expanding queries based on term-based concepts outperforms the standard VSM and outperforms or is equivalent to the pseudo relevance feedback (except at the NPL collection) and the results are not obtained by chance.

Additionally, the combination of our method with the pseudo relevance feedback outperforms the VSM and the pseudo-relevance feedback. Thus the term-based concepts are capable to improve the PRF just by adding the learned term weights.

7 Conclusions and Outlook

We have described a new approach for bridging the gap of different terminology within the user query and the searched documents by using term-based concepts. Each term of the query corresponds to a concept which is learned from the documents given by the ground-truth of other users. The selection relies on the similarity between the query terms and the learned concept rather than on the similarity between the terms of the collection nor on collection-based or hand-made thesauri. This approach can be used to improve the retrieval of documents in all domains like Digital Libraries, Document Management Systems, WWW etc. The approach performs the better the more user queries (users) are involved.

Additionally, we combined our method with the pseudo relevance feedback by adding the term weights of our learned term-based concepts.

Our experiments made on five standard test collections with different sizes and different document types have shown considerable improvements vs. the original queries in the standard vector space model and vs. the pseudo relevance feedback (except at the NPL collection). The improvements seem to be not depending on the type nor the size of the collection and they are not obtained by chance.

In contrast to the relevance feedback, this approach is not relying on critical thresholds which are dangerous and mostly differ from collection to collection.

Furthermore, this approach can be perfectly used in search machines where new queries with their appropriate relevant (user-voted) documents can be easily added to the 'collection', for example in Digital Libraries, Document Management Systems or the WWW. These new queries can be used to build an increasing approach and for a constant learning of the stored concepts. The vital advantage is that each user can profit from the concepts learned by other users. The more queries are learned (by the same or by other users) the better our approach will perform.

Some experiments are planed to use user-voted relevance feedback instead of collection-given ground-truth to test the performance on 'real-life' data. Furthermore, it is planed to make some experiments on the influence of the weights ω_t for each term.

An approach on passage-based retrieval by Kise [7] has shown good improvements vs. LSI and Density Distribution. An interesting idea for the future is not using the complete relevant documents for expanding the query and not using the N top ranked terms but using terms of relevant passages within the documents. With this idea just the relevant passages are used to learn the concepts. This should increase the quality of the expanded queries and we will be able to do a further evaluation of each concept in great detail, i.e. on the term level.

8 Acknowledgements

This work was supported by the German Ministry for Education and Research, bmb+f (Grant: 01 IN 902 B8).

References

1. 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
2. Baeza-Yates R., Ribeiro-Neto B.: Modern Information Retrieval. Addison-Wesley Pub. Co., 1999. ISBN 020139829X
3. ftp://ftp.cs.cornell.edu/pub/smart/
4. http://trec.nist.gov/
5. Hull D.: Using Statistical Testing in the Evaluation of Retrieval Experiments. In Proceedings of the 16th Annual Int. ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 329 - 338, 1993
6. Jansen B.J., Spink A., Bateman J. and Saracevic T.: Real Life Information Retrieval: A Study of User Queries on the Web, In SIGIR Forum, Vol. 31, pp. 5-17, 1988
7. Kise K., Junker M., Dengel A., Matsumoto K.: Passage-Based Document Retrieval as a Tool for Text Mining with User's Information Needs, In Proceedings of the 4th Internatl. Conference of Discovery Science, pp. 155-169, Washington, DC, USA, November 2001
8. Manning C.D. and Schütze H.: Foundations of Statistical Natural Language Processing, MIT Press, 1999
9. McCune B.P., Tong R.M., Dean J.S., Shapiro D.G.: RUBIC: A System for Rule-Based Information Retrieval, IEEE Transaction on Software Engineering, Vol. SE-11, No.9, September 1985
10. Minker J., Wilson, G.A. Zimmerman, B.H.: An evaluation of query expansion by the addition of clustered terms for a document retrieval system, Information Storage and Retrieval, vol. 8(6), pp. 329-348, 1972
11. Peat H.J., Willet, P.: The limitations of term co-occurrence data for query expansion in document retrieval systems, Journal of the ASIS, vol. 42(5), pp. 378-383, 1991
12. 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 Tamperensis 672. ISBN 951-44-4582-1; ISSN 1455-1616. June 1999
13. Qiu Y.: ISIR: an integrated system for information retrieval, In Proceedings of 14th IR Colloqium, British Computer Society, Lancaster, 1992
14. Salton G., Buckley C.: Term weighting approaches in automatic text retrieval. Information Processing & Management 24(5), pp. 513 - 523, 1988
15. Sparck-Jones K.: Notes and references on early classification work. In SIGIR Forum, vol. 25(1), pp. 10-17, 1991
16. Smeaton A.F., van Rijsbergen C.J.: The retrieval effects of query expansion on a feedback document retrieval system. The Computer Journal, vol. 26(3), pp. 239 – 246, 1983
17. Yang Y. and Liu X.: A Re-Examination of Text Categorization Methods. In Proceedings of the 22nd Annual Int. ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 42 - 49, Berkeley, CA, August 1999