TCL - An Approach for Learning Meanings of Queries in Information Retrieval Systems

Stefan Klink, Armin Hust, and Markus Junker
German Research Center for Artificial Intelligence (DFKI GmbH)
P.O. Box 2080, 67608 Kaiserslautern, Germany
{stefan.klink, armin.hust, markus.junker}@dfki.de

May 29, 2002

Abstract

The accuracy of ad-hoc document retrieval systems has plateaued in the last years. At DFKI, we are working on so-called collaborative information retrieval (CIR) systems which unintrusively learn from their users search processes. For a first step towards techniques, we focus on a restricted setting in CIR in which only old queries and correct answer documents to these queries are available for improving on a new query. For this restricted setting we propose a method called term-based concept learning (TCL) which learns conceptual description terms occurring in queries. A new query is then interpreted using the previously learned concepts. Combined with pseudo-relevance feedback, TCL shows very encouraging results.

1 Introduction

Information Retrieval (IR) Systems have been studied in Computer Science for decades. The traditional ad-hoc task in Information Retrieval is to find all documents relevant for an ad-hoc given query. Much work has been spent into improving this task, in particular in the Text Retrieval Evaluation Conference series (TREC) [7]. In 1998, it was decided on TREC-8 that this task should not be longer persuaded within TREC, in particular because the accuracy has plateaued in the last years. At DFKI, we are working on approaches for Collaborative Information Retrieval (CIR) which learn to improve retrieval effectiveness from the interaction of different users with the retrieval engine. Such systems may have the potential to overcome the current plateau in ad-hoc retrieval.

Figure 1 illustrates the general scenario of CIR. Typically, a document retrieval system is used by many users. A typical search in a retrieval system consists

of several query formulations: often, the answer documents to the first query do not directly satisfy the user. Instead, he has to reformulate its query taking the found answer documents into consideration. Such refinement may consist of specializations as well as generalizations of previous queries. In general, satisfying an information need requires to go through a search process with many decisions on query reformulations. The idea of CIR is to store these search processes as well as ratings of found answer documents (if available) in an archive. Subsequent users with similar interests and queries should then benefit from knowledge automatically acquired by the CIR system based on the stored search processes. This should result in shorter search processes and better retrieval quality for subsequent users.

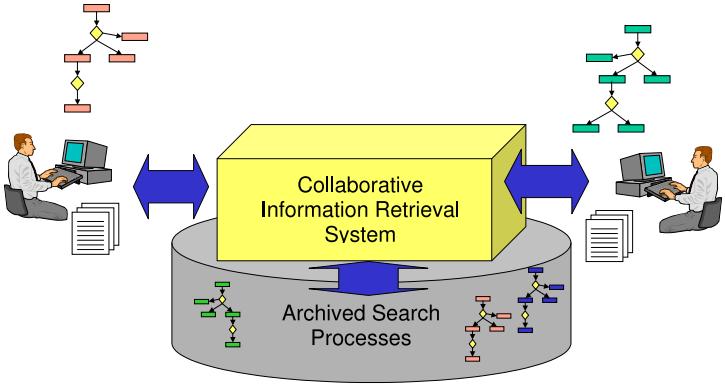

Figure 1: Scenario of Collaborative Information Retrieval.

In this paper we focus on the investigation of a simple CIR scenario. Given a number of old queries people asked a search engine and their corresponding answer documents, we try to improve on an arbitrary new query. In section 2 we first introduce two well-known retrieval models we rely on: the vector space model and the pseudo relevance feedback model. In section 3 we propose a CIR approach called TCL. This approach decomposes the new query into individual terms and enriches each term by concepts learned from the former queries and their answer documents. We also propose two approaches for combining TCL with pseudo relevance feedback Section 4 describes the experimental results obtained with the TCL method and compares them with the vector space model and the standard pseudo relevance feedback model. In section 5 we briefly review our new method.

2 The vector space model and pseudo relevance feedback

The basic retrieval model we rely on is the vector space model (VSM)[1], [4]. In this model documents as well as queries are represented by vectors. The relevance of a document with respect to a query is mapped to a similarity function between the query vector and all document vectors. More formally, documents as well as queries are represented by vectors $(w_1, w_2, ..., w_m)$. Each position i in the vectors corresponds to a specific term t_i in the collection. The value w_i indicates the weighted presence or absence of the respective term in the document or query. For weighting we rely on a variant of the standard tf-idf weighting schema [1]. In this weighting schema a word is weighted higher in a document/query if it occurs more often in the document/query. It is also weighted higher if the word is rare in the document collection. The similarity sim between a given query q and a document d is computed by $sim(d,q) = \frac{d \cdot q}{\|d\| \cdot \|q\|}$ where $\| \cdot \|$ is the Euclidean norm of a vector. For retrieving all documents to a given query, all documents of the underlying collection are ranked according to their similarity to the query and the top-ranked documents are given to the user.

It is immediately clear that documents may be relevant to query even if they do not share any word withe the query. Unfortunately, the standard VSM will always return zero similarity in this case. So-called query expansion techniques do overcome this problem by expanding the user given query q to a new enriched query q' which is then used in the standard VSM. Pseudo relevance feedback (PRF) is a very well known query expansion technique. It enriches the original query q by the terms of the top-ranked documents with respect to q. We are using a variation of PRF described in [3]: Let E be the set of document vectors given by

$$E = \left\{ d_j \left| \frac{sim(d_j, q_k)}{\max_{1 \leq i \leq N}\{sim(d_i, q_k)\}} \geq \theta, \quad 1 \leq j \leq N \right. \right\} \tag{1}$$

where q_k is the original query and θ is a similarity threshold. The expanded query vector q' is obtained by

$$q' = q + \alpha \frac{r}{\| r \|}, \text{ with } r = \sum_{r_j \in E} r_j \tag{2}$$

Note that the PRF approach has two parameters, θ and α.

3 Term-based Concept Learning

We first present the basic term-based concept learning (TCL) method (section 3.1). We then introduce two methods of combining TCL with pseudo relevance feedback (section 3.2).

3.1 Basic Method

Retrieval with short queries is much harder as compared to retrieval with long queries. This is because shorter queries often provide less information for retrieval. 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 the 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 users information need, although they do contain the same keyword as the query.

Another problem 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 or the same terminology used in the documents, respectively.

In order to solve these problems our TCL approach maps each individual term in a user's query to a representation of the concept it stands for. The mapping is learned by previous queries and their answer documents.

The TCL method is based on the learning of concepts for each term occurring within the current query. It is divided into two phases: the learning phase and the expansion phase:

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 documents 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 ro enrich the new query with the concept.

For the formal description of the learning phase, we need the following abbreviations:

- $\mathbf{D} = \{d_1, \ldots, d_N\}$: the set of all documents.
- $\mathbf{Q} = \{q_1, \ldots, q_L\}$: the set of all known queries.
- $q_k = (w_{1k}, ..., w_{ik}, ..., w_{Mk})^T$ represented within the vector space model (see section 2). For each term of the query the appropriate weight w_{ik} is between 0 and 1.
- $\mathbf{R}^+(q_k) = \{d_j \in \mathbf{D} \mid d_j \text{ is relevant for } q_k\}$.

Now, in the first step of the learning phase for each term w_i the set Q_i of all queries containing term w_i is computed:

$$\mathbf{Q}_i = \{(w_{1k}, ..., w_{ik}, ..., w_{Mk})^T \in \mathbf{Q} \mid w_{ik} \neq 0\} \tag{3}$$

If the i-th term doesn't occur in any query q_k then $\mathbf{Q}_i$ is empty.

In the second step all documents relevant to any query of $\mathbf{Q}_i$ are collected in the set $\mathbf{D}_{ik}$:

$$\mathbf{D}_{ik} = \{d_j \mid d_j \in \mathbf{R}^+(q_k) \wedge q_k \in \mathbf{Q}_i\} \tag{4}$$

In the last step of the learning phase the concept of i-th term is build as the sum of all documents which are relevant to the (previous) queries which have the term in common:

$$C_i = \sum_{d_j \in \mathbf{D}_{ik}} d_j \tag{5}$$

As queries and documents, a term-based concept is represented by a vector. If no query q_k contains term i, the corresponding concept C_i is represented by $(0, .., 0)^T$.

In the expansion phase the learned concepts are applied to a new user's query $\tilde{q} = (\tilde{w}_1, ..., \tilde{w}_i, ..., \tilde{w}_M)^T$ as follows:

$$\tilde{q}' = \tilde{q} + \sum_{i=1}^{M} \omega_i C_i \tag{6}$$

where the ω_i are parameters for weighting the corresponding concept. In the experiments described below ω_i is set to 1 and can be ignored.

Before applying the expanded query, it is normalized by

$$\tilde{q}'' = \frac{\tilde{q}'}{\parallel \tilde{q}' \parallel} \tag{7}$$

3.2 Combination with Pseudo-Relevance Feedback

In modern classification systems good results are obtained by combining several methods together. Thus, we combined the TCL method with the PRF in two ways: First, we applied the PRF and TCL method in parallel with the users query and combined the resulting documents of both methods by adding them to the query. Second, we applied TCL to expand the query with the concept terms. After this, the expanded query is applied with the PRF method.

3.2.1 Parallel combination

This method combines for each query the expansion resulting from PRF and the expansion resulting from the TCL method by adding them with a weighting factor. The query is build by (cmp. (2,6)):

$$\tilde{q}' = \tilde{q} + \beta r + \sum_{i=1}^{M} \omega_i C_i \tag{8}$$

Before applying the expanded query, it is normalized with equation (7). This method is reported as the PRF+TCL method.

3.2.2 Sequential combination

The sequential combination continues the idea of the PRF with an additionally feedback step which is done by the TCL method in advance. Thus, in the first step TCL is applied with the user's query and the query is expanded with (6). After that, PRF is applied with the TCL expanded query (cf. (2)). This method is reported as the PRF(TCL) method.

4 Experimental Evaluation

The new methods were evaluated on eight document collections from Information Retrieval. Section 4.1 describes the experimental setup while section 4.2 presents the results.

4.1 Setup

For our experiments we use standard Information Retrieval test collections as provided by [6] and [7]. These collections were originally made for testing the effectiveness of different IR systems on a wide range of (artificial) queries. The collections offer the advantage that for all queries the correct answer documents are known. On the other hand the queries are very likely not typical for real word document retrieval system. In particular, as opposed to real world systems, we expect little overlap in query topics. This may hurt the performance of our approaches.

In the experiments we used the following eight collections:

- the CACM (titles and abstracts from the journal 'Communications of the ACM'), CISI (Institute of Scientific Information) and CRAN (aeronautics abstracts) collections are available at [6]. All collections are provided with queries and their ground truth.

	CACM	CISI	CRAN	CR desc	CR narr	CR title	FR	AP90'
size(MB)	1.2	1.4	1.4	93	93	93	69	3.7
# documents	3204	1460	1400	27922	27922	27922	19860	723
# diff. terms	3029	5755	2882	45717	45717	45717	50866	17502
mean doc. length	18.4 (short)	38.2 (med)	49.8 (med)	188.2 (long)	188.2 (long)	188.2 (long)	189.7 (long)	201.8 (long)
# queries	52	112	225	34	34	34	112	353
mean query length	9.3 (med)	23.3 (long)	8.5 (med)	7.2 (med)	22.8 (long)	2.9 (short)	9.2 (med)	3.2 (short)
mean of relev. docs/query	15.3 (med)	27.8 (high)	8.2 (med)	24.8 (high)	24.8 (high)	24.8 (high)	8.4 (med)	2.8 (low)

Table 1: Statistics about test collections

- the CR (congressional record) collection. The CR collection is contained in the TREC test collections disk 4 [7], accompanied by the ground truth for 34 selected queries out of the TREC standard queries 251 - 300. We created three test cases for the CR collection, using the TREC queries of different length in order to investigate the influence of query length. The "CR-title" contains the "title" queries (the shortest query representation), the "CR-desc" contains the "description" queries (the medium length query representation), the "CR-narr" contains the "narrative" queries (the longest query representation).

- the FR (federal register) collection. The FR collection is contained in the TREC test collections disk 2, accompanied by the ground truth for 112 selected queries out of the TREC standard queries 51 - 300.

- a modified version AP90' based on the AP90 (associated press articles) collection contained in the TREC test collections (disk 3). Originally the AP90 collection contains 78321 documents. From the TREC-9 Question Answering track (QA) we selected the question set 201-893. Questions 201-700 were created without reference to the document set. Then in a separate pass equivalent but re-worded questions (questions 701-893) were created from a subset of these 500 questions [8]. Because of the method used for the construction of this set (especially questions 701 - 893) we expected to get higher similarities between different questions. From the ground truth data provided with the QA-track we selected only those questions having a relevant answer document in the AP90 document collection. Thus we reduced our test data to 723 documents and 353 questions.

Table 1 lists some statistics about the collections we used after stemming and stopword elimination has been carried out.

The evaluation of the new learning methods (TCL, PRF+TCL, PRF(TCL)) was done using a leave-one-out technique: From the set of queries $Q = \{q_1, \ldots q_n\}$

in a collection we selected each $q_k \in Q$ and extended it based on the queries $Q \setminus q_k$. Effectiveness was measured using macro-averaged recall/precision and averaged precision. Details on these measures can be found in [1]. In order to identify significant differences among methods we use the macro-averaged t-test (see, e.g., [9]) on the averaged precision (error probability $\alpha = 0.05$).

4.2 Results

Some of the methods we compared contain parameters. For all methods with parameter (VMS, PRF+TCL, PRF(TCL)) we first searched for the parameter setting maximizing the averaged precision for a collection. Table 4.2 shows the respective parameter values. In the TCL approach, the values ω_i were not optimized but set to 1.

		CACM	CISI	CRAN	CR desc	CR narr	CR title	FR	AP90'
PRF	α	1.7	0.7	1.3	0.5	0.4	0.6	0.6	0.2
	θ	0.35	0.7	0.9	0.85	0.95	0.75	0.55	0.75
PRF+TCL	α, θ	as in PRF above							
	β	0.41	0.62	1.06	0.84	3.58	0.57	0.09	0.72
PRF(TCL)	α	1.9	0.2	0.4	0.3	0.0	0.6	0.3	0.1
	θ	0.70	0.85	0.90	0.95	0.00	0.85	0.45	0.95

Table 2: Best parameter values for methods PRF, PRF+TCL, and PRF(TCL)

In this section the results of the experiments are presented. Recall/precision graphs were generated according to Figure 4.2 shows the recall/precision graphs for the test collections. Each graph contains the results for the methods VSM, PRF, TCL, PRF+TCL, and PRF(TCL).

Table 4.2 shows a comparison of the new methods using the t-test on the macro-averaged precision (The corresponding values of the averaged precision are listed in table 4. Full recall precision graphs are displayed in figure 2.). Each method was tested against each other. The entries in the table can be interpreted as follows:

- $+$: average precision of X is better than of Y.
- $-$: average precision of X is worse than of Y.
- $\circ$: no statement on differences can be made on the given error level.

Table 4 shows that PRF outperforms the VSM in all cases while for TCL the comparison with VSM does not allow a clear statement: In three collections it performs better than the VSM and in four collections it performs worse. The comparison of PRF+TCL and PRF also does not allow a clear statement: In one collection PRF+TCL performed better while in one other collection it performed worse. The methods PRF(TCL) showed best results of all: In three collections

X	Y	CACM	CISI	CRAN	CR desc	CR narr	CR title	FR	AP90'
PRF	VSM	+	+	+	+	+	+	+	+
TCL	VSM	+	–	–	∘	–	+	+	–
PRF+TCL	PRF	+	∘	∘	∘	∘	–	∘	∘
PRF(TCL)	PRF	+	∘	∘	∘	–	∘	+	+

Table 3: Results of macro t-test

	CACM	CISI	CRAN	CR desc	CR narr	CR title	FR	AP90'
VSM	13.0	12.0	38.4	17.5	17.3	13.5	8.5	74.3
PRF	19.9	12.9	43.5	20.4	19.2	16.9	18.1	75.5
TCL	28.2	10.0	34.2	16.7	8.2	16.0	14.0	62.9
PRF(TCL)	30.4	12.7	42.6	20.6	17.3	18.0	19.9	77.3
PRF+TCL	30.8	12.6	44.4	19.4	15.7	18.1	15.0	76.2

Table 4: Average precision obtained in different methods in percent

PRF(TCL) was significantly better than PRF. In four collection it did at least not hurt the effectiveness significantly. Unfortunately, in the CR-narr collection, it turned out to be significantly worse than PRF. This could be explained by the exceptionally long queries in this collection (22.8 words in average): Many words in a query may imply many words which are not directly related to the queries content. If many irrelevant words are extended to concepts, this may lead to noise in the new query. Fortunately, query lengths as observed in the CR-narr collection do not play a major role in practice.

5 Summary

We have proposed new methods for query expansion which rely on previously observed queries and the corresponding answer documents. A basic method called TCL was combined with pseudo relevance feedback in two different ways. The combined method PRF(TCL) has significantly higher averaged precision in 3 collections with typical query lengths.

It has to be noted that the collections we were using are standard IR collection. They do not reflect a query distribution as on a real world IR system in which we expect a much higher overlap between query contents and thus better results of our methods.

6 Acknowledgements

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

References

[1] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison-Wesley Publishing Company, 1999.

[2] D. Hull. Using statistical testing in the evaluation of retrieval experiments. In *Proceedings of SIGIR-93*, pages 329–338, 1993.

[3] K. Kise, M. Junker, A. Dengel, and K. Matsumoto. Experimental evaluation of passage-based document retrieval. In *Proceedings of the Sixth International Conference on Document Analysis and Recognition (ICDAR'01)*, 2001.

[4] C. Manning and H. Schütze. *Foundations of Natural Language Processing*. MIT Press, 1999.

[5] Y. Qiu and H.-P. Frei. Concept-based query expansion. In *Proceedings of SIGIR-93, 16th ACM International Conference on Research and Development in Information Retrieval*, pages 160–169, Pittsburgh, US, 1993.

[6] `ftp://ftp.cs.cornell.edu/pub/smart`.

[7] `http://trec.nist.gov`.

[8] E. M. Voorhees and D. Harman. Overview of the ninth text retrieval conference (TREC-9). In *Proceedings of the Ninth Text Retrieval Conference (TREC-9)*, 2001.

[9] Y. Yang and X. Liu. A re-examination of text categorization methods. In *Proceedings of SIGIR-99*, pages 42–49, 1999.

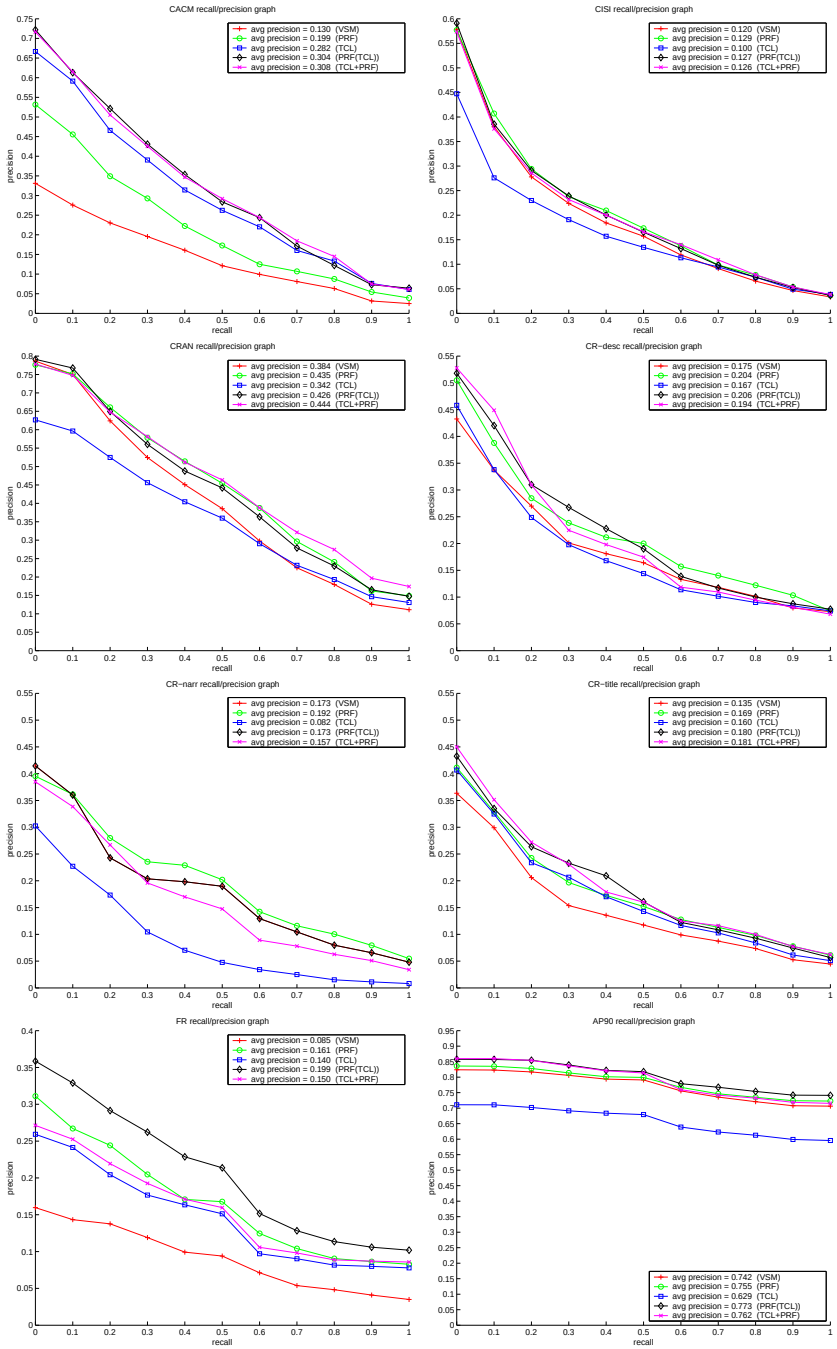

Figure 2: Recall/precision graphs.