Improving a Basic Retrieval Method by Links and Passage Level Evidence

Daniel Knaus, Elke Mittendorf, Peter Schäuble
(knaus|mittendorf|schauble)@inf.ethz.ch
Swiss Federal Institute of Technology (ETH)
CH-8092 Zürich, Switzerland

Abstract

A conventional text retrieval method is improved by two additional sources of evidence of relevance: first, by a passage retrieval method that is based on Hidden Markov Models and second, by a so-called link method which was originally developed for hypertext retrieval. The results show that the two additional sources of evidence improve a conventional vector space retrieval method both in a consistent and in a complementary way. The link method has the ability to retrieve relevant documents whose description vectors are not very similar to the query description vector whereas the passage retrieval method has the ability to improve an existing ordering of the documents.

1 Introduction

Our approach is aimed at improving a conventional text retrieval method by means of two additional sources of evidence of relevance. The conventional retrieval method is called the *basic method* B. It consists of a well-known weighting scheme which is sometimes referred to as *lnc.ltc* (Knaus & Schäuble, 1993).

Passage retrieval is used as a second source of evidence of relevance. We used our *passage retrieval method* P which is based on Hidden Markov Models (Mittendorf & Schäuble, 1994). This approach has the advantage that the passages are not restricted to a fixed length. Furthermore, the parameters can be estimated automatically by means of the Baum–Welch reestimation formula.

A so-called *link method* L is used as a third source of evidence of relevance. The link method was originally developed for hypertext retrieval (Frei & Stieger, 1995). Since the TREC documents are not linked, we automatically create appropriate links before applying the hypertext retrieval method.

We combined the three sources of evidence of relevance in an ad-hoc way, i.e. by trying several combinations to find the best combination. When combining several sources of evidence of relevance the scores of the particular sources have to be normalized. We circumvented the normalization problem by using rank numbers. We believe that the problem of combining several sources of evidence of relevance needs further investigations.

2 Multiple Levels of Evidence

2.1 Basic Method

TREC-2 has shown that using a general stoplist, a word normalization algorithm, and an appropriate weighting scheme already leads to fairly good results in terms of retrieval effectiveness (Knaus & Schäuble, 1993). Our basic method for TREC-3 uses the stoplist of the SMART System (Ide & Salton, 1971) consisting of 571 stopwords. The remaining words are normalized by Porter's suffix stripping algorithm (Porter, 1980). Thus, our indexing features consist of Porter reduced words which do not occur in the SMART stoplist. The analysis of all documents of the document collection provides the feature frequencies $\mathrm{ff}(\varphi_i, d_j)$ and the document frequencies $\mathrm{df}(\varphi_i)$. The feature frequency $\mathrm{ff}(\varphi_i, d_j)$ denotes the number of occurrences of the indexing feature φ_i within the document d_j and the document frequency $\mathrm{df}(\varphi_i)$ denotes the number of documents containing the feature φ_i. It should be noticed that the document frequencies are always determined by disk 1 and disk 2 (D1D2) independent of whether they are used for the ad-hoc or the routing task. Hence, query features not occurring in D1D2 have a document frequency $\mathrm{df}(\varphi) = 0$ even though they may occur in disk 3 (D3).

The retrieval function is equivalent to the cosine measure. The indexing method uses *lnc.ltn* feature weighting. This weighting scheme showed the best retrieval effectiveness in our experiments at TREC-2 and it has the advantage that features of routing documents not occurring in the training collection are independent of collection statistics such as inverse document frequencies. Throughout the paper we refer to this method as the basic method B.

query q
documents d_j, $0 \leq j < n$
indexing features φ_i, $0 \leq i < m$
feature frequencies $\mathrm{ff}(\varphi_i, d_j)$, $\mathrm{ff}(\varphi_i, q)$
document frequencies $\mathrm{df}(\varphi_i)$
normalized inverse document frequencies
$\mathrm{nidf}(\varphi_i) := 1 - \frac{\log(\mathrm{df}(\varphi_i)+1)}{\log(n+1)}$
query feature weights (ltn weighted)
$b_i := (1 + \log(\mathrm{ff}(\varphi_i, q))) * \mathrm{nidf}(\varphi_i)$
document feature weights (lnc weighted)
$a_{ij} := (1 + \log(\mathrm{ff}(\varphi_i, d_j)))$
document lengths
$\mid \vec{d_j} \mid := \sqrt{\sum_{i:\varphi_i \in d_j} {a_{ij}}^2}$
Retrieval Status Values
$RSV_B(q, d_j) := \frac{1}{|\vec{d_j}|} * \sum_{i:\varphi_i \in q, d_j} a_{ij} * b_i$

2.2 Link Method

The link method is based on Stieger's hypertext retrieval method (Frei & Stieger, 1995). Hypertext retrieval methods do not only take into account the relevance of a node but also the relevance of its neighbors (Frisse, 1988). The main ideas of Stieger's hypertext retrieval method are the following.

1. We take into account the relevance of a link with respect to the query. Below we will elaborate on what we mean by the relevance of a link.
2. We reduce the hypertext to a substructure consisting only of links that are sufficiently relevant to the query.
3. We reduce the variation of how much the retrieval status value of a node is affected by the retrieval status values of its neighbors.

Since the TREC data does not contain links between the documents we first generate links automatically before applying Stieger's hypertext retrieval method.

We generate a link from document d_j to document d_k if they contain a common phrase λ which satisfies the following three conditions.

1. The common phrase $\lambda = \{\varphi_h, \varphi_i\}$ consists of two indexing features φ_h and φ_i which must occur in both documents d_j and d_k in such a way that only short non-stop words may occur in between where a word is considered as short if it consists of one or two letters. Examples are "exploration of the universe" which becomes "EXPLOR UNIVERS" or "flight spectrometer" which becomes "FLIGHT SPECTROMET".
2. In addition, the two words φ_h and φ_i which constitute the link between d_j and d_k must occur sufficiently many times.
$$\begin{aligned} \mathrm{ff}_{min} &\leq \min\{\mathrm{ff}(\varphi_h, d_j), \mathrm{ff}(\varphi_i, d_j)\} \\ \mathrm{ff}_{min} &\leq \min\{\mathrm{ff}(\varphi_h, d_k), \mathrm{ff}(\varphi_i, d_k)\} \end{aligned}$$
In this way, we require that the phrase is an important phrase in both documents d_j and d_k.
3. Finally, the document frequency of the phrase λ is within a certain interval.
$$\mathrm{df}_{min} \leq \mathrm{df}(\lambda) \leq \mathrm{df}_{max}$$

This is in accordance with Luhn's ideas for selecting good indexing features (Luhn, 1958). By adding such links to the document collection we obtain a hypertext to which the link method is applied. Subsequently, a link is identified by the corresponding phrase $\lambda = \{\varphi_h, \varphi_i\}$.

In what follows, we describe the hypertext retrieval method. First, the hypertext is reduced to a substructure consisting only of links that are sufficiently relevant to the query. In particular, we obtain the corresponding substructure by restricting ourselves to those links which satisfy the following requirements.

1. The two reduced words φ_h and φ_i constituting the link λ must occur in the query.
2. The relevance of the link $\lambda = \{\varphi_h, \varphi_i\}$ with respect to the query q must be above the threshold c,
$$RSV_{lnk}(q, \lambda) \quad > \quad c,$$
where the relevance is estimated by
$$RSV_{lnk}(q, \lambda) \quad := \quad \frac{\sum_{i:\varphi_i \in \lambda} b_i \mathrm{nidf}(\varphi_i)}{\sqrt{\sum_{i:\varphi_i \in q} b_i^2}}.$$

Using the estimated relevance of the links, the neighborhood $D_j(q)$ of a document d_j with respect to the query q is defined by

$$(d_k, \lambda) \in D_j(q) \quad :\Longleftrightarrow \quad \begin{cases} d_k \in D1D2 \\ d_j \xrightarrow{\lambda} d_k \\ \lambda \subseteq q \\ RSV_{lnk}(q, \lambda) > c \end{cases}$$

where $d_j \xrightarrow{\lambda} d_k$ means that a link λ from d_j to d_k exists which satisfies the three conditions given above. Finally, the retrieval status values of the link method L are defined by

$$RSV_L(q, d_j) := \frac{1}{\mid D_j(q) \mid} \sum_{(d_k,\lambda) \in D_j(q)} RSV_{lnk}(q, \lambda) RSV_B(q, d_k)$$

parameter	value
ff_{min}	2
df_{min}	3
df_{max}	7000

Table 1: Parameters.

where the parameters of the link method we have chosen are shown in Table 1. Because the parameter c was adapted to the different tasks its values are shown in Section 3.

2.3 Passage Level Evidence

In this section we describe how relevant passages are extracted from documents and how the documents are scored using the scores of the passages. Hidden Markov Models (HMM) and the Viterbi-algorithm provide a natural method for retrieving passages without knowing anything about the structure of documents and without assuming anything about the format and the size of the passages. Document and passage retrieval based on Hidden Markov Models are described in (Mittendorf & Schäuble, 1994).

For our purposes we roughly assume that—with respect to a query—each document can be segmented into three passages: an irrelevant passage, followed by a relevant passage, again followed by an irrelevant passage. The HMM which models these assumptions is visualized in Figure 1. An HMM can be regarded as a set of

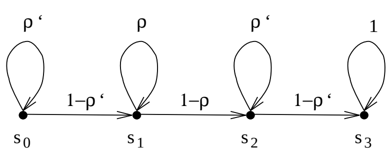

Figure 1: HMM for modeling a document

stochastic production rules for documents. These can be described by a stochastic syntax diagram. So in our case a document is produced while traversing the syntax diagram in Figure 1. Upon each arrival in a state a token is generated according to a probability distribution that belongs to this state. The next state is chosen according to the transition probabilities ρ and ρ'. In the states s_0 and s_2, tokens are generated which belong to irrelevant passages, in the state s_1, tokens are generated which belong to a relevant passage. In the state s_3 and only in this state, a token is generated that marks the end of the document.

A document is considered as a sequence of tokens. To gain query independence and a higher stability of the model parameters a description step is introduced: Each token of the sequence is mapped to a value that describes the similarity between the token and the query. If the token is an occurrence of the feature φ_i, we used

$$\mathrm{sim}(\varphi_i, q) := \mathrm{ff}(\varphi_i, q) \cdot \mathrm{nidf}^2(\varphi_i)$$

as the corresponding similarity value. Thus, the HMM can be considered as a probabilistic production rule for sequences of real values that describe the similarity of a token and a given query q.

The output probabilities are adjusted automatically by the Baum–Welch algorithm (Baum, 1972). We used the queries 1–50 and D1D2 for training. For the distribution of the similarity values in the states s_0 and s_2 we use irrelevant query-document pairs as input for the Baum–Welch algorithm, and for the state s_1 we use relevant query-document pairs. The transition probabilities ρ and ρ' are set to 0.5.

Given a document d_j, a query q, and the corresponding sequence of similarity values, the Viterbi-algorithm performs a probabilistic parse through the given HMM. It determines a path through the model which has a maximal probability of producing the given sequence of similarity values. There is a subsequence that is produced in the state s_1 while traversing the HMM along this path. The passage $p_j(q)$ which belongs to this subsequence is a subsequence of the document d_j. We will call this passage the best passage of d_j with respect to q.

The retrieval status value of a document is the similarity between the query and the best passage determined by the Viterbi-algorithm. The feature weighting is implemented as follows:

$$a'_{ij} \quad := \quad 1 + \log(\mathrm{ff}(\varphi_i, p_j(q))).$$

Since passages from relevant documents are likely to be longer than passages from irrelevant documents, it is not reasonable to normalize, as this is done for example for method B, therefore the inner vector product is chosen to be the retrieval function, i.e.

$$RSV_P(q, d_j) := \sum_i a'_{ij} b_i.$$

This passage retrieval based method is referenced to as method P.

2.4 Combination

In this section we describe how the basic method B, the link method L and the passage method P are combined. There are several possibilities for combining two sets of retrieval status values to a new set of retrieval status values. Given a method X and a method Y they can be combined by using a linear combination of the corresponding retrieval status values. The resulting combination method is denoted by XY. The basic method

B and the link method L are combined in this way:

$$RSV_{BL}(q,d_j) := \alpha RSV_B(q,d_j) + (1-\alpha)RSV_L(q,d_j).$$

Table 2 and Table 4 show which values are chosen for parameter α.

For this kind of combination the retrieval status values of two different methods have to be on the same scale. If they are not, they could be normalized by e.g. the median or a maximal value. But sometimes one method might be rather on a logarithmic scale and the other on a quadratic scale or some scale not easy to recognize. In these cases a very sophisticated normalization is required.

Another possibility of combination is the combination of the rank numbers instead of the combination of retrieval status values (Bartell et al., 1994). In this way normalization can be avoided but some information will be lost. For a given method X and a given method Y the new method that is derived by the combination of rank numbers is denoted by $X \circ Y$. Since the retrieval status values of the method BL and of the method P are not on a comparable scale, this kind of combination was used to combine the lists of ranked documents derived from RSV_{BL} and RSV_P.

If the document d_j is ranked on position r_{BL} with respect to RSV_{BL} and if the passage p_j which belongs to d_j is ranked on position r_P with respect to RSV_P, then

$$RSV_{BL\circ P}(q,d_j) := \beta(-r_{BL}(q,d_j)) + (1-\beta)(-r_P(q,d_j)).$$

For efficiency reasons only those documents are analyzed that are retrieved either by the method L ore by the method B, i.e.

$$RSV_P(q,d_j) := \begin{cases} \sum_i a'_{ij} b_i, & \text{if} \quad r_B(q,d_j) \leq 1000 \\ & \text{or} \quad r_L(q,d_j) \leq 1000. \\ 0 & \text{otherwise.} \end{cases}$$

Also for efficiency reasons, $r_{BL}(q,d_j)$ was assumed to be 1001 if d_j was retrieved by P and not retrieved by BL (i.e. rank greater than 1000). Similar, $r_P(q,d_j)$ was assumed to be 1001 if d_j was retrieved by BL but not by P.

If a document does not appear among the top 1000 documents on the list of ranked documents of the methods BL or P, its rank number r_{BL} respectively r_P is assumed to be 1001. The value β was chosen according to Table 2.

3 Results

We submitted the results of two ad-hoc experiments (ETH001 and ETH002) and the results of two routing experiments (ETH003 and ETH004). ETH001 and ETH003 were determined by linear combination of the retrieval status values obtained from the basic method B and from the link method L. ETH002 and ETH004 were determined by combining the ranks of the documents obtained from the combination BL and from the passage retrieval method P. An overview of all *ad-hoc* experiments is given in Table 2. Table 3 shows the retrieval effectiveness of these experiments. The numbers within brackets are comparisons to the basic method B. The following facts can be derived from Table 3:

method	description and parameter values
RSV_B	basic method nidf derived from D1D2
RSV_L	link method links derived from D1D2 $c = 0.1$
RSV_P	passage retrieval method HMMs trained using topics 1-50 and D1D2
RSV_{BL}	linear combination of RSVs from methods RSV_B and RSV_L $\alpha = 0.4$
$RSV_{B\circ L}$	linear combination of ranks from methods RSV_B and RSV_L $\beta = 0.83$
$RSV_{B\circ P}$	linear combination of ranks from methods RSV_B and RSV_P $\beta = 0.47$
$RSV_{BL\circ P}$	linear combination of ranks from methods RSV_{BL} and RSV_P $\beta = 0.8$

Table 2: Ad-hoc experiments (151-200 versus D1D2).

- Method L is much less effective than method B but 510 new relevant documents are found that have not been retrieved by the basic method B (i.e. that are not among the 1000 top ranked documents).
- Comparing the most effective BL combination ($\alpha = 0.6$) and the most effective $B \circ L$ combination ($\beta = 0.1$) shows that combining retrieval status values is cleary more effective than combining ranks. The optimal α and the optimal β have been determined using TREC-2 data.
- The combinations perform much better than the methods B, L, or P perform on their own.
- The improvements of the simple combinations BL and $B \circ P$ are complementary. The overall combination $BL \circ P$ has a better retrieval effectiveness than both simple combinations.

retrieval method	avg. precision	exact R-precision	# rel. retr. (max. 9805)	# rel. not retr. by B	submitted as
RSV_B	0.2578	0.3247	6033	-	-
RSV_L	0.1045 (-59.5%)	0.1800 (-44.6%)	3071 (-2962)	510	-
RSV_P	0.1833 (-28.9%)	0.2557 (-21.3%)	6087 (+54)	396	-
RSV_{BL}	0.2737 (+6.3%)	0.3374 (+3.9%)	6202 (+169)	364	ETH001
$RSV_{B \circ L}$	0.2616 (+1.2%)	0.3217 (-0.9%)	6134 (+101)	324	-
$RSV_{B \circ P}$	0.2853 (+10.7%)	0.3435 (+5.8%)	6156 (+123)	330	-
$RSV_{BL \circ P}$	0.2916 (+13.1%)	0.3475 (+7.0%)	6237 (+204)	381	ETH002

Table 3: Retrieval effectiveness of the ad-hoc experiments.

method	description and parameter values
RSV_B	basic method nidf determined from D1D2
RSV_L	link method links determined from D1D2 $c = 0.075$
RSV_P	passage retrieval method HMMs trained using topics 1-50 and D1D2
RSV_{BL}	linear combination of RSVs from methods RSV_B and RSV_L $\alpha = 0.3$
$RSV_{BL \circ P}$	linear combination of ranks from methods RSV_{BL} and RSV_P $\beta = 0.8$

Table 4: Routing experiments (101-150 versus D3).

retrieval method	avg. precision	exact R-precision	# rel. retr. (max. 9353)	submitted as
RSV_B	0.2993	0.3460	6024	-
RSV_{BL}	0.3092 (+3.3%)	0.3440 (-0.1%)	6242 (+218)	ETH003
$RSV_{BL \circ P}$	0.3154 (+5.4%)	0.3468 (+0.2%)	6295 (+271)	ETH004

Table 5: Retrieval effectiveness of the routing experiments.

The same methods have been applied to the *routing* experiments. We did not take advantage of the relevance assessments that were available for the routing topics. The parameter values used in the routing experiments are given in Table 4. Table 5 summarizes the results. The same statements can be derived as from the ad-hoc experiments but the improvements are somewhat smaller.

4 Conclusions

The results of our experiments seem to indicate that the link method and the passage retrieval method improve conventional vector space retrieval in a *complementary* way. The link method has the ability to retrieve relevant documents whose description vectors are not very similar to the query description vector. On the other hand, the passage retrieval method has the ability to improve an existing ordering of the documents. At this moment it is not clear whether it is possible (desirable) to measure how complementary two retrieval methods are. Finally, we lack a theoretical framework to combine several retrieval methods. The current adhoc approach to find an optimal combination is rather time consuming. There may exist more efficient ways to optimize combinations of retrieval methods.

For TREC-3, we concentrated on the ad-hoc experiments. In the routing task, we did not use the relevance information of the training collection. Thus, our methods could be improved by applying a simple relevance feedback method or by modifying the neighborhood $D_j(q)$ of the link method L.

Acknowledgements: The authors would like to thank Roger Aeppli for the implementation of the link method, for running most of the experiments and for his support in writing this paper. We would also like to thank Rene Notter for implementing the passage retrieval method.

References

Bartell, B. T., Cottrell, G. W., & Belew, R. K. (1994). Automatic Combination of Multiple Ranked Retrieval Systems. In *ACM SIGIR Conference on R&D in Information Retrieval*, pp. 173–181.

Baum, L. E. (1972). An Inequality and Associated Maximization Technique in Statistical Estimation of Probabilistic Functions of Markov Processes. *Inequalities*, 3,1–8.

Frei, H. P., & Stieger, D. (1995). The Use of Semantic Links in Hypertext Information Retrieval. *Information Processing & Management*, 31(1),1–14.

Frisse, M. E. (1988). Searching for Information in a Hypertext Medical Handbook. *Communications of the ACM*, 31(7),880–886.

Ide, E., & Salton, G. (1971). Interactive Search Strategies and Dynamic File Organization. In Salton, G., editor, *The SMART Retrieval System—Experiments in Automatic Document Processing*, chapter 18. Prentice-Hall Inc.

Knaus, D., & Schäuble, P. (1993). Effective and Efficient Retrieval from Large and Dynamic Document Collections. In *TREC-2 Proceedings*, pp. 163–170.

Luhn, H. P. (1958). The automatic creation of literature abstracts. *IBM Systems Journal*, 2,159–165.

Mittendorf, E., & Schäuble, P. (1994). Document and Passage Retrieval Based on Hidden Markov Models. In *ACM SIGIR Conference on R&D in Information Retrieval*, pp. 318–327.

Porter, M. F. (1980). An Algorithm for Suffix Stripping. *Program*, 14(3),130–137.