データベースシステム　124－16
情 報 学 基 礎　62－16
（２００１．５．２２）

適合フィードバックを利用したクウェリ拡張における単語選択方法

真野博子 小川泰嗣
（株）リコー ソフトウェア研究所
{mano,yogawa}@src.ricoh.co.jp

Abstract

適合フィードバックを利用したクウェリ拡張において、追加すべき単語を選択するには、候補となっている単語の価値を評価する必要がある。これには、単語の重みを主に利用する方法が知られているが、この方法が、フィードバックに用いた文書が実際には適合文書とはかぎらない場合にも最適の方法であるかについて、これまであまり検討されることがなかった。本稿では、単語の重みを利用した単語評価が単語選択に与える影響について論じ、単語の重みを利用しない方法と比較する。

Term selection in automatic query expansion using pseudo-relevance feedback

Hiroko Mano and Yasushi Ogawa
Software Research Center, RICOH Co., Ltd.

Abstract

In automatic query expansion, expansion terms to be added to the original query are selected according to some term evaluation measure. Typically, to evaluate each term, a combination of term weight and other term statistics is used, whether the feedback documents from which the terms are taken are actually relevant or not. However, we found that the effectiveness of this method greatly depended on the actual relevance of the feedback documents and, when the relevance was limited, an approach not using term weight was more appropriate.

1. Introduction

In document retrieval, user's information requests are presented to the system in the form of a set of terms the user expects to find in the relevant documents. Very often, however, the terms used in the user-supplied query do not match the terms actually used in the relevant documents, due to possible different wording.

Query expansion is a technique whereby a user-supplied query is augmented, or expanded with additional terms taken from top-ranked documents returned by search using the query. The original idea was to have the user judge each top-ranked document whether it is relevant or not and use the information as relevance feedback to automatically determine which terms to add. However, concern about the extra burden on the user involved in the relevance judgement lead to introduction of pseudo-relevance feedback, with which all the top-ranked documents are assumed to be relevant and all the bottom-ranked documents are assumed to be non-relevant, without relevance judgement. Query expansion, with or without

user input, is known to improve overall retrieval effectiveness and has been successfully used in various retrieval systems [1, 6].

Our probabilistic-model-based retrieval system [4] is one such example and the following illustrates how retrieval with automatic query expansion using pseudo-relevance feedback proceeds in our system:

1. Query construction

   The query constructor accepts each topic, extracts words in each of the appropriate fields and constructs a query to be supplied to the ranking system. (See Appendix for more information on the query construction process.)

2. Initial retrieval

   The constructed query is fed into the ranking system, which then, scores each document and turns up a set of top-ranked documents as relevant to the topic (pseudo-relevant documents).

   The document score is, in essence, a sum of within-document term frequency (the number of occurrences of the term within the document), each weighted by an inverse of document frequency (the number of documents that contain the term). Within-document term frequency is used to capture the term's representativeness within each document and inverse document frequency is used to capture the term's distinctiveness within the target document set.

3. Query expansion

   The query expander collects and ranks the terms in the pseudo-relevant documents in the order of their appropriateness as expansion terms. The words ranked the highest are added to the original query, with the terms already in the query re-assigned new term weight.

4. Final retrieval

   The ranking system performs final retrieval using the modified query.

One key issue in automatic query expansion is how each term is evaluated for its appropriateness as an expansion term. The idea is that the better the term relates to the topic, the higher we rate the term. Classic term evaluation statistics, such as *term weight*, based on some form of inverse document frequency and what we call in this paper *term prevalence*, typically based on within-document term frequency, can be and have been adopted for this purpose.

An example of evaluation measures for expansion term expansion, or Term Selection Value, is, in essence:

$$TSV\text{-}1 = term\ weight \cdot term\ prevalence$$

in which weighty and prevalent terms get high scores.

Another possible evaluation measure is:

$$TSV\text{-}2 = term\ prevalence$$

in which only prevalent terms are valued.

The TSV-1 method was successfully used in our past experiments. However, recent experiments showed, under some circumstances — namely, when the top-ranked documents, assumed to be relevant, turn out to be non-relevant — the TSV-2 method was a better measure to gather additional terms that contribute to retrieval effectiveness. In this paper, we examine these two methods, that is, the one that uses term weight and the one that does not, using an experimental condition taking into account the number of relevant documents and compare each method's strengths and weaknesses.

2. More on Term Weight and Term Selection Value

2.1. Term weight

As mentioned above, the most often used measure of term's usefulness to guide search in a document set is term weight. In this section, we give a brief review of what term weight does and how it is calculated in the probabilistic model, using the well-known Okapi BSS model [6] as an example.

In the probabilistic model, term weight is based on the estimation of relative distribution of the term in relevant documents and non-relevant documents, on the premise that terms that appear only in relevant documents would be the most effective to distinguish between relevant documents and non-relevant documents.

Suppose we have R documents that are known to be relevant and S documents that are known to be non-relevant. Using the relevance weighting theory of the probabilistic model, we would be able to use the relevance information to assign weight to terms as:

$$w_t = \log \frac{r+0.5}{R-r+0.5} - \log \frac{s+0.5}{S-s+0.5}$$

where r is the number of relevant documents containing the term and s is the number of non-relevant documents containing the term. (0.5 is for mathematical adjustment only.) The weighting would rate higher those terms that appear in more of the relevant documents and less of the non-relevant documents.

On the other hand, when documents are retrieved with no prior information on relevance judgements, as in initial retrieval using a user-supplied query, term weight has to be calculated without R, S, etc.; hence we simply use the number of documents that contain the term and the number of documents that do not, as:

$$w_t = k_4 + \log \frac{N}{N-n} - \log \frac{n}{N-n}$$

where N is the number of documents in the collection and n is the number of documents in which the term occurs. (k_4 is a parameter.) This is essentially the same as:

$$w_t = \log \frac{N}{n}$$

or what is called inverse document frequency in general. With the absence of relevance information, the assumption here is that the smaller the number of documents that contain the term, the more effective the term would be to distinguish between relevant documents and non-relevant documents. This is term weighting used for document scoring in initial retrieval.

The two term weight functions, with and without relevance information, can be combined into one as:

$$\begin{aligned} w_t = {} & \frac{k_5}{k_5+\sqrt{R}}(k_4 + \log \frac{N}{N-n}) + \frac{\sqrt{R}}{k_5+\sqrt{R}} \log \frac{r+0.5}{R-r+0.5} \\ & - \frac{k_6}{k_6+\sqrt{S}} \log \frac{n}{N-n} - \frac{\sqrt{S}}{k_6+\sqrt{S}} \log \frac{s+0.5}{S-s+0.5} \end{aligned} \quad (1)$$

which is a weighted average of term weight calculated fully independent of relevance feedback and term weight calculated solely from relevance feedback. The size of R and S is taken into account to determine the balance, reflecting that the relevant document set and the non-relevant document set actually used are just small samples. (k_5 and k_6 are parameters.) This form of term weighting is used for document scoring in final retrieval after relevance feedback, to weight terms in the expanded query.

2.2. Term Selection Value

The term weight function (1) is not for document scoring only, however. It has been used, in Okapi and ours as well, as part of Term Selection Value, an evaluation measure to determine the value of a term as an expansion term. We call this type of TSV, which incorporates term weight, as TSV-1 in this paper.

The TSV-1 function we have used [4] is:

$$TSV\text{-}1 = w_t \cdot prev_t$$

in which the term weight is calculated by:

$$\begin{aligned} w_t = {} & \frac{k_5}{k_5+\sqrt{R}} \log \left(k_4' \frac{N}{N-n} + \frac{n}{N-n} \right) + \frac{\sqrt{R}}{k_5+\sqrt{R}} \log \frac{r+0.5}{R-r+0.5} \\ & - \frac{k_6}{k_6+\sqrt{S}} \log \frac{n}{N-n} - \frac{\sqrt{S}}{k_6+\sqrt{S}} \log \frac{s+0.5}{S-s+0.5}, \end{aligned}$$

and the term prevalence is calculated by:

$$prev_t = \left(\frac{\sum_{d \in R} \frac{f_{t,d}}{K + f_{t,d}}}{R} - \beta \frac{\sum_{d \in S} \frac{f_{t,d}}{K + f_{t,d}}}{S} \right),$$

$$K = k_1 \left((1 - b) + b \frac{l_d}{l_{ave}} \right),$$

where where $f_{t,d}$ is the within-document frequency of the term, l_d is the document length, l_{ave} is the average document length, and k_1, b and β are parameters. (The term weight calculation is slightly different from that of Okapi's for reasons of parameter adjustment.)

In the past experiments using TREC-7 and TREC-8 (See the following section for details on TREC), query expansion using the TSV-1 function above always resulted in retrieval effectiveness better than that of no query expansion, at least when averaged over all queries. Admittedly, for some queries, query expansion did not improve or even hurt performance, but overall, query expansion had been considered a certain way to boost retrieval effectiveness in our experiments.

The TREC-9 experiments, however, showed this was a false assumption; the result of runs with query expansion was far lower in retrieval effectiveness than that of runs without query expansion. For example, in one condition, the TSV-1 run achieved only 93% in average precision compared to the no-expansion run.

A cursory look at the selected terms after query expansion suggested term weight in TSV-1 might be causing the problem; the TSV-1 function seemed to favor too many of terms that are too rare that appear to be unrelated to the topic. However, follow-up experiments with different parameter values, especially those for term weight, failed to produce any gain. With little prospect of improvement in parameter tuning, we shifted the focus and began to test an alternative, TSV-2, dropping term weight.

$$TSV\text{-}2 = prev_t$$

The reasoning was that by just looking at term prevalence, or how often the term occurs in a relevant document, some of the rarer terms would be more likely to be left out since they tend to appear a fewer times per document. Subsequent experiments showed that, with TREC-9, using TSV-2 instead of TSV-1 was indeed more effective in improving performance.

One reason that TSV-1 did poorly in TREC-9 compared with TSV-2 was, it seemed, that the top-ranked documents initially retrieved contained a relatively fewer relevant documents. Also, there seemed to be some topics that were suited for TSV-1 but not for TSV-2, and vice versa. In the following, we describe the experiment we conducted to investigate these issues and some of the characteristics of these two TSVs we found in the experiment.

3. Experiment

3.1. TREC evaluation environment

In this experiment, we used the TREC-8 Web Track topics and its WT2g document collection and the TREC-9 Web Track topics and its WT10g document collection. These are materials offered in the Eighth and the Ninth Text REtrieval Conference, where retrieval effectiveness of participating retrieval systems is evaluated within a same setting [7].

In each of the TREC-8 and TREC-9 topic sets, 50 topics are given. Each TREC topic consists of three fields: title, description and narrative. The title field is a brief statement of the topic of interest in a few words. The description field elaborates the topic in a longer sentence; the narrative field gives some examples of what is relevant, and occasionally, what is not. Here is a sample topic taken from TREC-9:

<title> What is the composition of zirconium?

<desc> Find documents that describe the physical properties of zirconium.

<narr> A document is relevant if it describes the element itself or its behavior under various physical conditions. A document is not relevant if it gives only the uses of zirconium or its compounds without

stating why it combines with certain elements.

The WT2g document collection and the WT10g document collection are subsets of the VLC2 collection, a 100GB snapshot of the WWW [3]; the WT2g is a 2GB subset consisting of about 250,000 documents and the WT10g, 10GB, 1,700,000 documents.

For relevance judgement, a list of documents officially judged relevant is supplied. Using the list, documents returned by the retrieval system are evaluated for retrieval effectiveness, typically using average precision, which calculates the proportion of retrieved documents that are relevant, as measured after each relevant document is retrieved and averaged.

3.2. Experimental conditions

Obviously, the quality of query expansion may well be influenced by the quality of the initially retrieved top-ranked documents, in particular, how many of them are actually relevant, and our experimental goal was to investigate whether the degree of influence varies depending on the use of term weight when selecting expansion terms.

In the experiment, we created three conditions of top-ranked documents: worst-possible, best-possible and realistic, each consisting of at most 10 documents. To simulate the worst case scenario where initial retrieval fails to turn up any relevant document, a set of non-relevant documents (NONE set) was constructed by weeding out all relevant documents from top-ranked documents that contain query terms while retaining the relative rank of each non-relevant document that will remain in the set. Similarly, a set of relevant documents (ALL set) was created by picking up only relevant documents from the top-ranked documents.

For comparison purposes, a set of top-ranked documents as they were obtained in initial retrieval without modification (SOME set) was also prepared. In the TREC-8 runs, approximately 48 to 50% of the set were relevant documents on the average. In the TREC-9 runs, the ratios of relevant documents were in the range of approximately 25 to 35%.

We tested two term selection measures, TSV-1 and TSV-2 in Section 3.2 to select expansion terms. S was set to 0 in our experiments. In addition, each TSV has some sort of cut-off measure, TSV-1 eliminates terms with small r that appear in the same set of documents as previously selected terms and TSV-2 eliminates terms with low weight.

Queries were created using either the title field only or the title and desc fields.

4. Result and analysis

4.1. Mean average precision

The experimental runs resulted in the mean average precision measurements in Tables 1 and 2.

The result shows that, with the ALL set, selecting terms using TSV-1 resulted in the higher average precision than selecting terms using TSV-2. On the other hand, with the NONE set, using TSV-1 affected retrieval more adversely than using TSV-2. The outcomes using the SOME set also suggest that the more the top-ranked document set contains relevant documents, the greater TSV-1 outperformed TSV-2. Note also that using original queries with no expansion produced better results in all runs with the NONE set,

Table 1. TREC-8 topics and data collection

		title only	title+desc
ALL	TSV-1	0.4575	0.4571
	TSV-2	0.4050	0.4130
SOME	TSV-1	0.3503	0.3687
	TSV-2	0.3423	0.3603
NONE	TSV-1	0.2848	0.3091
	TSV-2	0.3088	0.3327
No expansion		0.3247	0.3420

Table 2. TREC-9 topics and data collection

		title only	title+desc
ALL	TSV-1	0.3578	0.3870
	TSV-2	0.3122	0.3545
SOME	TSV-1	0.2021	0.2427
	TSV-2	0.2171	0.2725
NONE	TSV-1	0.1686	0.1866
	TSV-2	0.1850	0.2406
No expansion		0.2073	0.2608

but with the SOME set, only in the runs using TSV-1 in TREC-9 runs. This implies that there is at least one form of query expansion that is effective when around one third or so of top-ranked documents is relevant.

Table 3 illustrates what portion of the relevant documents are affected most by query expansion using TSV-1 and TSV-2; that is, where in ranking query expansion placed the top-ranked relevant documents and the residual relevant documents relative to the non-relevant documents in final retrieval. The table lists, for the title-and-desc run using TREC-9, average precision for top-ranked relevant documents and residual relevant documents returned by the run using SOME set and the no-expansion run. From this, we see poorer performance registered for the top-ranked documents in the run using TSV-1. This implies that term selection based on TSV-1 leads to lowered ranks of the relevant documents originally in the feedback set, when there is not a sufficient number of them.

Table 3. SOME set and no expansion

	Top-ranked	Residual
TSV-1	0.5703	0.1280
TSV-2	0.6390	0.1276
No expansion	0.6279	0.1159

The same statistics for the ALL set is in Table 4. (Remember that the number of relevant documents in SOME set is different from that in ALL set, hence no direct comparison is to be made between the two.) With the ALL set, TSV-1's performance is superior to that of TSV-2 for both the top-ranked documents and the residual documents.

Table 4. ALL set

	Top-ranked	Residual
TSV-1	0.6452	0.1440
TSV-2	0.5643	0.1244

4.2. Term and topic analysis

Comparing the terms selected, we notice those selected by TSV-1 resulted in the average document frequency considerably lower than that of those selected by TSV-2, regardless of which set the terms came from. For instance, in the TREC-9 desc-and-title runs using the ALL sets, the average document frequency of the terms selected by TSV-2 is 44368, while that of the terms selected by TSV-1 is 21929, less than half the frequency resulting from TSV-2. This indicates that the TSV-1 tends to consistently favor uncommon terms over common terms. In other words, with TSV-1, term weight exerts a strong influence on term selection, despite the fact that the function also takes term prevalence into account.

When we look at the result topic by topic, we get a more detailed picture of how these two TSVs affect query expansion and consequently final retrieval effectiveness, depending on each topic. In the following,

we examine what we call expansion effect, the difference of average precision between expansion and no expansion, again in the title-and-desc runs using TREC-9.

The first thing we notice is that even with the ALL set, some topics did not benefit from query expansion at all; for 7 topics using TSV-1 and 4 topics using TSV-2, query expansion had no or negative effect on retrieval effectiveness even when all documents used for relevance feedback were truly relevant. Most of them already enjoy relatively high precision at high recall points even without query expansion, suggesting that these are the topics for which there are no definitely better terms than the original ones that express the topic's intent.

Conversely, even with the NONE set, query expansion was of some benefit to some of the topics, although the benefit is quite small; for 4 topics using TSV-1 and 10 topics using TSV-2. It indicates top-ranked non-relevant documents are not entirely useless and may serve as a source of additional terms if used properly.

For the majority of the topics, however, the ALL set results in a hike and the NONE set, a drop in average precision. With the ALL set, there are topics that benefited greatly from query expansion, and for these topics, the effect is greater with TSV-1 than with TSV-2. From the terms selected by TSV-1 for these topics, the TSV-1's strength seems to come from its ability to draw in more specific terms that help to narrow the search scope not explicitly stated in the query.

Take, for instance, Topic 470, which states "Identify documents that discuss beneficial uses of mistletoe." It would not be difficult to find documents that mention mistletoe, but to choose only those that describe "beneficial uses" would be a challenge. For this, TSV-1 was able to pick up some names of herbs, e.g., fenugreek and lobelia from the ALL set, which gives an example of health benefits from other medicinal plants.

Use of TSV-2 seems to work in the opposite direction. With NONE set, there are topics for which runs using TSV-2 yielded comparatively better, or less poor, results than TSV-1. The overall impression is that terms selected by TSV-2 tend to be supplementary terms that help to widen the search scope already defined by the original query but with limited influence.

Topic 455 is an example in which TSV-2 resulted in gain and TSV-1 resulted in loss, using the NONE set. The requirement was to "Find documents that indicate when Jackie Robinson made his major league debut," and both TSV-1 and TSV-2 found some related terms, such as "baseball" and "Negro," but only TSV-2 was able to ward off such terms as "Clockers," a name of a movie that is hardly related to the topic but weighs heavily because of its low document frequency.

5. Conclusion

The experiment showed that term selection using term weight was sensitive to how many of the top-ranked documents were actually relevant and that, with this method of term selection, the range of fluctuation caused by the ratio of relevant documents was quite wide. On the other hand, it was found that term selection not using term weight produced relatively stable result.

This suggests, when reliable relevance feedback is available, such as in an interactive environment, including term weight in its term selection evaluation would be very effective, while under most realistic circumstances, dropping term weight in term selection seems to produce more stable results.

References

[1] C. Buckley, G. Salton and J. Allan. The effect of adding relevance information in a relevance feedback environment. In *Proc. of 17th ACM SIGIR Conf.*, 1994.

[2] C. Fox. A stop list for general text. *ACM SIGIR Forum*, Vol. 24, No. 2, pp. 19–35, 1991.

[3] D. Hawking, E. Voorhees, N. Craswell and P. Bailey. Overview of the TREC-8 Web Track. In *The Eighth Text REtrieval Conference (TREC-8)*, 2000.

[4] Y. Ogawa, H. Mano, M. Narita, and S. Honma. Structuring and expanding queries in the probabilistic model. In *The Eighth Text REtrieval Conference (TREC-8)*, 2000.

[5] S.E. Robertson and S. Walker. On relevance weights with little relevance information. In *Proc. of 20th ACM SIGIR Conf.*, pp. 16–24, 1997.

[6] S. Walker, S.E. Robertson, M. Boughanem, G.J.F. Jones and K. Sparck Jones. Okapi at TREC-6. In *The Sixth Text REtrieval Conference (TREC-6)*, 1998.

[7] E. Voorhees and D. Harman. Overview of the Eighth Text REtrieval Conference (TREC-8). In *The Eighth Text REtrieval Conference (TREC-8)*, 2000.

Appendix

The following is a brief description of how queries are constructed from topic statements. For details, see [4].

From text in each topic, stemmed word tokens are extracted and all terms but stopwords are put together into a query. Fox's word list [2] was used to define the stopwords.

```
<title> industrial waste disposal
#OR(industrial,waste,disposal)
```

Also extracted from original text are noun phrases, identified by LT_CHUNK developed at the Edinburgh Language Technology Group, and the words in those phrases form term pairs to be incorporated into the query with the proximity and the order specified.

```
#WINDOW[1,1,ordered](waste,disposal)
#WINDOW[1,1,ordered](industrial,waste)
#WINDOW[2,50,unordered](waste,disposal)
#WINDOW[2,50,ordered](industrial,waste)
```

Some adjustments on term weight are additionally specified in the query, to indicate the importance of the phrases relative to the single words and to take into account the importance of the topic fields where the terms come from.

```
#OR(industrial,waste,disposal,
#SCALE[0.4](#WINDOW[1,1,ordered](waste,disposal)),
#SCALE[0.4](#WINDOW[1,1,ordered](industrial,waste)),
#SCALE[0.25](#WINDOW[2,50,unordered](waste,disposal)),
#SCALE[0.25](#WINDOW[2,50,ordered](industrial,waste)))
```