# Document Expansion versus Query Expansion for Ad-hoc Retrieval

*Bodo Billerbeck* *Justin Zobel*

School of Computer Science and Information Technology
RMIT University, GPO Box 2476V, Melbourne, Australia
{bodob,jz}@cs.rmit.edu.au

November 18, 2005

**Abstract** *In document information retrieval, the terminology given by a user may not match the terminology of a relevant document. Query expansion seeks to address this mismatch; it can significantly increase effectiveness, but is slow and resource-intensive. We investigate the use of document expansion as an alternative, in which documents are augmented with related terms extracted from the corpus during indexing, and the overheads at query time are small. We propose and explore a range of corpus-based document expansion techniques and compare them to corpus-based query expansion on TREC data. These experiments show that document expansion delivers at best limited benefits, while query expansion – including standard techniques and efficient approaches described in recent work – delivers consistent gains. We conclude that document expansion is unpromising, but it is likely that the efficiency of query expansion can be further improved.*

**Keywords** Document expansion, automatic query expansion, pseudo relevance feedback, efficiency

## 1 Introduction

Word mismatch is a common problem in information retrieval. Most retrieval systems match documents and queries on a syntactic level, that is, the underlying assumption is that relevant documents contain exactly those terms that a user chooses for the query. However, a relevant document might not contain the query words as given by the user. Query expansion (QE) is intended to address this issue. Other topical terms are located in the corpus or an external resource and are appended to the original query, in the hope of finding documents that do not contain any of the query terms or of re-ranking documents that contain some query terms but have not scored highly.

A disadvantage of QE is the inherent inefficiency of reformulating a query. With the exception of our earlier work [2], these inefficiencies have largely not been investigated. In this work we proposed improvements to the efficiency of QE by keeping a brief summary of each document in the collection in memory, so that during the expansion process no time-consuming disk accesses need to be made. While some of the methods proposed in this earlier research more or less maintain effectiveness, the process is sped up by roughly two-thirds. However, expanding queries using the best of these methods still takes significantly longer than evaluating queries without expansion.

In this paper, we explore the use of document expansion (DE) as an alternative to QE. In DE, documents are enriched with related terms. Although, while not prohibitively so, there is a significant cost associated with expanding documents; this is undertaken at indexing time, and there is only marginal cost at retrieval time. In principle it is reasonable to suppose that DE will help resolve the problem of vocabulary mismatch and thus yield benefits like those obtainable with QE.

We propose two new corpus-based methods for DE. The first method is based on adding terms to documents in a process that is analogous to QE: each document is run as a query and is subsequently augmented by expansion terms. The second method is based on regarding each vocabulary term as a query, which is expanded and used to rank documents. The original query term is then added to the top-ranked documents.

Our experiments measure the efficiency and effectiveness of QE and DE on several collections and query sets. We find that, on balance, DE leads to improvements in effectiveness, but few of the measured gains are statistically significant; the computational cost at query time is small. In contrast, both standard QE and the efficient QE that we proposed earlier [2] lead to gains in most cases, many of them significant, while the efficient QE is less than twice the cost of querying without expansion.

Our experiments were, within the constraints of our resources, reasonably exhaustive. We tested several alternative configurations of DE and explored the parameters, but did not observe useful gains in effectiveness. We conclude that corpus-based DE is unpromising for small sets of terms. We did not explore QE to the same extent, yet found effectiveness to consistently improve, and thus believe that further gains in performance may be available.

**Proceedings of the 10th Australasian Document Computing Symposium, Sydney, Australia, December 12, 2005.**

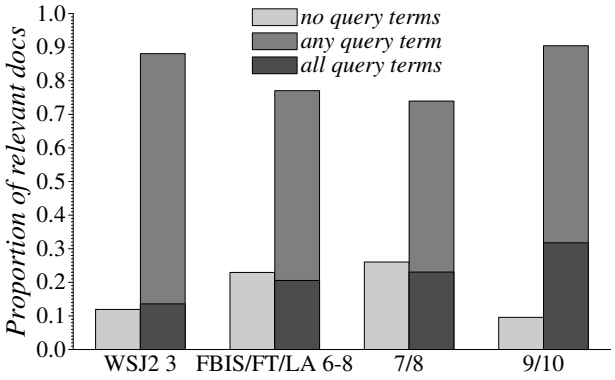


Figure 1: *The proportion of relevant documents that contain none, any, or all query terms over all title queries for each data set as shown (collections and queries are discussed in Section 4). Stopping, but no stemming, was used to generate this graph.*

## 2 Background

User queries often do not reflect the exact terminology of a document. Whereas a document might be on the exact topic of a query, this document will not be retrieved if it doesn't contain any of the key words in the user query. Figure 1 shows that as many as 25% of documents that are judged to be relevant do not contain *any* terms that appear in a concise query. The actual proportion might be much larger than this figure suggests, since in the TREC framework – from which the graph was produced – the relevance of only a relatively small number of documents is judged for each query. QE adds related terms to a query, so that those documents will be included in the ranking.

Early successful attempts of QE were based on relevance feedback [12]. Since this required the user to assess a large number of documents for their relevance, it proved to be impractical. Rather than asking users to assess whole documents, *interactive QE* suggests topical terms to the user that do not appear in their query. The user is then able to add any number of those terms to the query. Since users are generally reluctant to provide such information, and it was found that algorithms are just as likely as non-expert users to pick terms that enhance (or conversely, do not enhance) retrieval [13, 18], research has since shifted to *pseudo relevance feedback*. Terms, that are heuristically found to be related to the topic of the original query are automatically added to the query, without user intervention.

One approach to *automatic query expansion* methods – that require no user input other than the original query – is *global analysis* where collections are analysed using metrics such as term co-occurrences. Correlating terms are then used to build a thesaurus which is drawn on during query time by simply looking up related terms that are subsequently appended to the query.

*Local analysis* methods (such as that proposed by Robertson and Walker [11]) retrieve a set of documents through an initial ranking by the original query (see Algorithm 1 and Figure 2a). Terms from those documents

**Algorithm 1** Conventional QE through local analysis

1: run original query $q$ and rank docs in collection
2: select top 10 documents as local set $R$
3: extracted all terms $t$ from local set $R$
4: **for all** terms $t \in R$ **do**
5: calculate term selection value
6: **end for**
7: rank terms $t$ based on their a term selection value
8: add top $|E|$ terms to the original query
9: run expanded query $q'$ and rank documents

are retrieved. The frequency of those terms amongst the set of retrieved documents as well as collection statistics are taken into account in order to determine which terms are added to the query.

While global analysis mechanisms are inherently much more efficient than local ones (only dictionary lookups are performed during query time, rather than costly document retrieval and parsing), they are also likely to be less successful [19]. The difference in effectiveness is based on the problem that a term can take on different meanings, depending on which context it appears. Local analysis methods inherently disambiguate word senses better, as expansion terms are sourced from documents that are retrieved with the whole query, rather than individual query terms. Because of this difference in performance and the fact that our methods proposed below are based on local analysis blind relevance feedback, we compare the effectiveness and efficiency of our proposed DE techniques to that of a standard local analysis technique.

## Improving QE efficiency

Local analysis QE consists of several steps, some of which are time-consuming. First, there is an initial ranking process, where documents are identified that are presumably on the topic of the query. Next those documents are retrieved. Since most queries will rank different documents, these documents are most likely not cached (assuming a reasonable amount of memory) and have to be fetched from disk at a significant penalty in time. This is the most costly subtask of the QE process. Once documents are in main memory, they have to be parsed and statistics of term occurrences in respect of the local set of documents have to be computed. At a relatively minor cost, statistics of those terms for the whole document collection have to be looked up. Terms are then chosen and appended to the query. Finally the query has to be re-run, which requires not only the re-processing of inverted lists for the original query terms, but new lists have to be retrieved, decoded, and analysed.

Only the first step needs to be performed in the absence of expansion. There is no previous research concerned with accelerating the QE process in information retrieval, apart from our earlier paper [2], where we use a summary of each document consisting of that

document's top *tf.idf* terms. During querying, a fixed number of terms – or alternatively, terms with a *tf.idf* value above a certain threshold – is kept in memory for each document. While performing local analysis, rather than retrieving documents from disk, the in-memory summaries are referenced. This procedure improves querying throughput by a factor of two, while effectiveness is only marginally degraded. Although they were able to avoid the time-consuming retrieval of documents from disk, they restricted their focus to standard approaches to QE.

## Document expansion

Whereas DE has recently been applied in various areas of information retrieval, it has not been used instead of QE to improve ranking effectiveness, with the exception of Ide and Salton [5]. While not actually expanding documents, Ide and Salton manipulate their vector representation not unlike the DE methods proposed in this paper, although – unlike in this paper – they use actual relevance feedback. They propose to change the document vector space so that relevant documents are closer to the query vector. They achieve improvements of 10% to 15%.

Actual DE (that is, not just manipulating document vectors, but actually adding terms to documents) was first used by Singhal and Pereira [15] in the context of speech retrieval. Since speech recognition is unreliable (at the time of publishing, Singhal and Pereira report error rates of up to 60% for particular collections – although speech recognition has improved since), transcribed documents are expanded with related terms from a side corpus. Singhal and Pereira achieve a relative increase in average precision of 12% in addition to employing pseudo relevance feedback based on the technique proposed by Rocchio [12].

*Latent semantic indexing* [4] is in effect a DE method, however for information retrieval it was found to be inferior to the vector space model [9].

Li and Meng [8] use DE for spoken document retrieval with good improvements in Cantonese monolingual retrieval and in Mandarin cross-language retrieval.

Both Lester and Williams [6] and Levow and Oard [7] have used DE for topic tracking. Whereas Lester and Williams use DE to enrich topic profiles and do not specify whether it bears any benefit, the latter get consistent improvements in Mandarin cross-lingual retrieval by expanding the documents to be tracked.

With the exception of Lester and Williams (who expand only translated documents), all other work mentioned above uses DE in the context of enriching possibly incorrectly translated documents.

## Query associations

One of our proposed DE methods (detailed in Section 3) is in essence quite similar to *query association* as used in the context of effective retrieval [14]. We describe these here and highlight the differences to our proposed method later. Scholer et al. make use of a query log, by running each query of the log and adding the text of the query to the top $N$ ranked documents. Each document is augmented with the top $M$ queries that achieved the highest similarity score. They found that good values for $M$ and $N$ are 19 and 39 respectively.

We previously made use of query associations in conjunction with QE [1] with good success, however, for that work we stored associations separately and then expanded queries from the especially created *surrogates* conventionally.

# 3 Document expansion methods

Rather than expanding a query from an initially retrieved set of documents, which is time-consuming, DE expands documents with potential query terms that occur in similar documents. While this expansion process is reasonably costly, it is done prior to indexing time. Query times are only slightly increased, since inverted lists are on average, say 10% longer, depending on which DE method is chosen.

There are several ways to expand documents. All methods have one aim: to eliminate inefficient run-time QE, while getting some effectiveness of a local analysis mechanism. Each of the following proposed methods makes use of local analysis at indexing time and expands the original documents with additional terms.

## Selection and weighting measures

Before describing the different DE techniques we propose, we first explain underlying equations needed to arrive at expanded documents. We use one similarity measure, three different measures to select expansion terms, and one measure that weights selected terms.

**Similarity measure.** To measure the similarity of queries to documents, we use Okapi BM25 [16] in all our experiments, where constants $k_1$ and $b$ are set to $1.2$ and $0.75$ respectively. We set $k_3$ to 0, motivated by the assumption that each term in contemporary queries [17] only occurs once.

**Term selection measures.** Depending on the expansion method, we use different measures to select terms from a set of candidate terms.

We use the *term selection value* [11] in our experiments for ranking terms, if not stated otherwise:

$$TSV_t = \left(\frac{f_t}{N}\right)^{f_{r,t}} \binom{|R|}{f_{r,t}}$$

where $f_t$ is the number of documents in the collection in which term $t$ occurs in, $N$ is the total number of documents in the collection, and $f_{r,t}$ is the number of the $|R|$ top ranked documents in which term $t$ occurs.

An alternative is the *Kullback-Leibler divergence*, which specifies the distance between two probability densities. In other words, each term in the local set of documents ($R$) gets a value associated with the relative rareness of a term in the current set as opposed to the whole collection. The *KLD* weight of terms that occur

**Algorithm 2** Document centric expansion

1: **for all** documents $d \in$ collection **do**
2: formulate query $q$ to consist of all terms $t$ in $d$
3: rank documents in collection against $q$
4: select top 10 docs (other than $d$) as local set $R$
5: using *TSV*, select top $|E| = 25$ terms from $R$ (excluding $t \in q \cap d$) and append to $d$
6: **end for**

relatively often (or seldom) in the local set in contrast to the entire collection will receive a higher (lower) value than terms that appear as often as their term frequency suggests. The *KLD* can be calculated as [3, page 154]:

$$KLD_t = \frac{f_{r,t}}{|R|} \times \log\left(\frac{f_{r,t}}{|R|} \times \frac{F + 0.01 \times |V|}{F_t + 0.01}\right)$$

where $F_t$ is the total number of occurrences of term $t$ in the collection, $F$ is the combined total number of occurrences of all terms in the collection, and $|V|$ is the number of unique terms in the collection.

**Term weighting.** In all our experiments, expansion terms are weighted[1] by the Robertson/Sparck Jones relevance weight [10], to be used in the Okapi formula:

$$rw_t = \frac{1}{3}\log\frac{(f_{r,t} + 0.5)(N - f_t - |R| + f_{r,t} + 0.5)}{(|R| - f_{r,t} + 0.5)(f_t - f_{r,t} + 0.5)}$$

## Document centric DE

For this DE technique each complete document is run as a query and the top $|E|$ expansion terms determined through local analysis are appended to the document (see also Algorithm 2 and Figure 2b). This method is conceptually similar to conventional QE. Although this way of expanding documents is reasonably time consuming, it could be sped up considerably by for instance using only the top $n$ *tf.idf* terms for each query.

Even though the Okapi variant that we use for our experiments is not well suited for queries with duplicate query terms, we found that using the standard BM25 formulation or the Cosine measure degrades results considerably. Using our training data, we found that selecting terms with the *KLD* worked consistently better than using their *tf.idf* value or *TSV*. Interestingly, it became clear that allowing terms which are already in a document to be appended to this document decreases effectiveness compared to restricting additions to new material. We also found that augmenting a document with 10% of the number of tokens in a document works best, rather than adding a fixed number of terms or using a global threshold value for the selection value of each candidate term. That is, a document that contains 100 words is augmented with 10 more words. A side effect of DE is therefore that document collections and associated indexes are roughly 10% longer than the original collection and indexes after expansion.

A potential problem with DE is that terms that are used for augmenting documents tend to be quite rare across the collection. Adding rare terms to documents means that, after expansion, those terms will be less rare, which will have an effect on retrieval performance.

[1] The dampening factor of 1/3 helps to prevent query drift. It was recommended by unpublished correspondence with the authors.

**Algorithm 3** DE based on vocabulary

1: **for all** words $t \in$ vocabulary $V$ **do**
2: form query $q$ from $t$
3: rank documents $d$ against $q$
4: select top 10 documents $R$ as local set
5: rank candidate terms using *TSV*
6: append top $|E| = 25$ terms to $q$, forming $q'$
7: rank 100 documents ($X$) against $q'$
8: **for all** documents $d$ in $X$ **do**
9: calculate $s$, the similarity score of $q'$ to $d$
10: save $t, d, s$ triplets
11: **end for**
12: **end for**
13: **for all** documents $d$ in collection **do**
14: select $0.1 \times |d|$ terms with highest $s$ and add to $d$
15: **end for**

## Term centric DE

This approach to document expansion mimics more closely a reversal of the conventional local analysis algorithm. Imagine a query that consists of one term only. The role of QE is to identify documents that are *about* this term, but do not necessarily include this term. This is done by adding terms to the query that co-occur with the query term within the local set. After expansion we therefore retrieve documents that do not contain the query term but that do contain expansion terms. DE inverts this scenario: it puts the query term into those documents that contain the expansion terms. This has the effect of adding potential query terms that are on the same topic as the document but are missing from it. In other words, Algorithm 3 ideally adds terms to a document that would have lead to the document being ranked if the term had been run as a single original term in an expanded query (see also Figure 2c). Our hypothesis is that this algorithm is a good match for queries consisting of single terms, however less so for the case of multi-term queries.

This expansion method is considerably faster than the document centric approach. However, a problem that does not occur with the document centric approach arises in a setting where the collection grows, such as the web. Since the basis for selecting expansion terms is changing with the addition of new documents, the terms previously chosen for a particular document might be sub-optimal. Furthermore, it is difficult to determine the best expansion terms for added documents, as those documents did not exist when the collection was originally ranked against terms. An – admittedly expansive – solution is to rerun the DE process after a certain number of documents have been added. Possible optimisations are outside the scope of this paper.

In our experiments, we rank 100 documents against an expanded term, although we found that ranking any

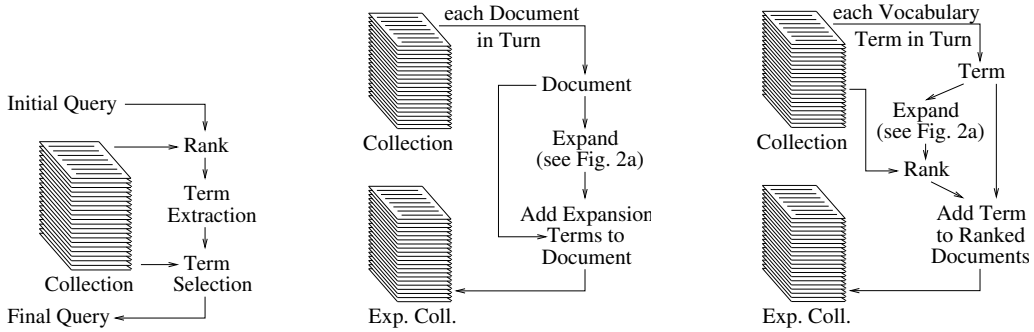


Figure 2: *The figure on the left (a) shows the central part of any expansion process proposed in this paper. The centre figure (b) shows how documents can be expanded by running each document as a query and adding the expansion terms back into the document. On the right (c) is shown how documents are expanded by running vocabulary terms as a queries and adding the terms to the top ranked documents.*

number between 90 and 110 documents works equally well. Contrary to the first method detailed above, we found that using the *TSV* to select terms worked better than choosing candidate terms based on their *KLD* values. We allowed terms to be added to documents even though they might already appear in that document. Surprisingly, we found that excluding terms on this basis leads to lower increases in effectiveness.

Query associations, as described in Section 2, differ from our DE methods in several ways. First, our techniques have the extra step of expanding a query with terms before ranking documents that a query gets associated with. Second, associations are based on external information in the form of query logs, whereas DE relies on within-collection data and statistics only. More importantly, the query association results [14] show that augmenting documents with queries works best when placing the restriction on queries to be associated with a document that all query terms must be present in the document. The effect of this restriction is that pertinent terms in a document get emphasised (their term count is increased and therefore the ranking of those documents is improved subsequently), rather than new terms – which address the problem of vocabulary mismatch – are added to the document. Adding new terms makes a document retrievable to queries that originally would not have ranked this document, even though it may be on the same topic.

## Expansion via phrases as queries

As an extension to the term centric expansion, instead of running individual vocabulary terms against the corpus to establish associations between those terms and documents, phrases can be used. This addresses a potential shortfall of the method above, which is a good match for queries consisting of single terms only.

We consider a phrase to consist of two or more contiguous terms that are not separated by either a stop word, an HTML or TREC tag, or any of the following characters: ?!,;:(){}[]. In separate experiments, we use maximal-length phrases and overlapping two-term phrases. As in the previous method, the phrases then get added to documents.

# 4 Experimental setup

We evaluate the proposed approaches in respect of effectiveness and efficiency as well as significance of results. As the underlying search engine we use Zettair.[2] We did not use stemming, but stopped queries. Although the local analysis parameters $|E|$ and $|R|$ are collection dependent, we did not tune those for each collection. Instead we use the default parameters of 25 and 10, respectively, in all cases.

**Test collections.** All our test data is drawn from the TREC conferences. To tune parameters and choose selection measures we used the Wall Street Journal articles from TREC disk 2, which covers issues from years 1990-92, referred to as WSJ2. With this collection we used the TREC 3 topics 151–200. We ran the title field as queries in all experiments to evaluate our system.

We used several collections to evaluate our techniques. One is sourced from the same TREC: Associated Press (AP); we used the AP data from disks 1 and 2 to match the TREC 3 topics and relevance judgements. We also used the newswire collection from TRECs 7 and 8 (NW). This collection is drawn from disks 4 and 5, without the congressional record. NW was used as a whole and also as several sub-collections from this collection, namely the Financial Times 1991-94 (FT), the Foreign Broadcast Information Service (FBIS) and the LA Times (LA). Testing was done against topic sets of TRECs 6, 7 and 8.

**Timings.** For timings, we used 100,000 stopped queries taken from two query logs collected for the Excite search engine [17]. Although these queries are web queries and not ideally suited to match the newswire data (we were not able to obtain a more suitable query log), these queries are adequate for testing the throughput only – rather than effectiveness – of the system.

Our timings were produced on two machines. The first is a Pentium IV 2.8 GHz machine with hyper-threading and 2 GB of main memory. The second is a dual Pentium III 866 MHz with 768 MB of main memory. In Table 1 these are denoted as *Lrg*

[2]Zettair is an open source search engine available from `http://www.seg.rmit.edu.au/`

| Expansion | WJS2 | | AP | | NW | | FBIS | | FT | | LA | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Method | Lrg | Ltl | Lrg | Ltl | Lrg | Ltl | Lrg | Ltl | Lrg | Ltl | Lrg | Ltl |
| None | 4.7 | 7.4 | 6.9 | 11.7 | 11.4 | 22.8 | 5.0 | 8.0 | 6.8 | 12.1 | 6.8 | 11.3 |
| QE | 25.4 | 47.1 | 29.0 | 52.5 | 145.9 | 211.2 | 49.4 | 123.5 | 41.2 | 87.6 | 32.6 | 62.3 |
| $S = 40$ | 7.5 | 14.8 | 11.4 | 22.5 | 20.9 | 52.0 | 8.1 | 16.3 | 11.8 | 24.4 | 10.6 | 20.6 |
| $Q = D$ | 5.3 | 8.6 | 7.7 | 13.3 | 12.1 | 24.8 | 5.5 | 9.4 | 7.6 | 13.7 | 7.3 | 13.3 |
| $Q \in V$ | 4.9 | 7.4 | 7.0 | 11.8 | 11.7 | 22.9 | 5.1 | 8.2 | 6.9 | 12.1 | 6.9 | 11.3 |
| $Q = P$ | 4.9 | 7.6 | – | – | – | – | – | – | – | – | – | – |
| $Q = B$ | 4.8 | 7.6 | – | – | – | – | – | – | – | – | – | – |

Table 1: *The efficiency of expansion techniques is shown as the average query time in milliseconds over 100,000 queries on a machine with a large amount of memory (Lrg) and one with little (Ltl).* None *specifies the baseline,* QE *shows the standard local analysis results, and* $S$=40 *shows the results for a summarisation technique.* $Q$=$D$ *is the document centric expansion technique.* $Q \in V$, $Q$=$P$, *and* $Q$=$B$ *are term centric and phrase centric approaches.*

and *Ltl* respectively. *Lrg* has ample amount of memory that easily fits – at least for the experiments with small collections – the whole document collection as well as inverted indexes and any major auxiliary data structures, such as document summaries, if applicable. Even though main memory was flushed before timings were commenced, eliminating any influence of caching from any previous timed runs, as all 100,000 queries are processed, all data is eventually cached. The effect of this to the conventional QE method is that the additional time requirement over the baseline is purely that of parsing documents, evaluating terms and processing a greater number of inverted lists, rather than the main cost associated with expanding queries from a local set in a typical environment, which is retrieving documents from disk. In practice, for larger collections, this scenario is unrealistic. We therefore also show timings for a machine that can fit only part of the collection and inverted lists in main memory.

**Significance testing.** One cannot assume that two result sets differ significantly from each other by simply observing the magnitude of the difference of an evaluation measure over a number of queries. To evaluate our results we make use of the non-parametric Wilcoxon matched-pairs signed ranks test since it places no assumption on the distribution of test data. In particular, a non-parametric test does not require that data is normally distributed, which is important for our purposes.

## 5 Results

Results are listed in Tables 1 and 2. Only methods that were successful on our training data are reported. The $S = 40$ rows in Tables 1 and 2 give results for one of the most successful methods we previously explored when using in-memory document summaries [2]. For this method each document is summarised by the top 40 *tf.idf* terms of that document. During query time, the summaries for all documents are kept in memory. The parameter of $S = 40$ was not tuned for the WSJ2 collection. The memory overheads for this method are as follows: WSJ2: 11.7 MB, AP: 26.1 MB, NW: 84.0 MB, FBIS: 20.8 MB, FT: 33.2 MB, and LA: 20.4 MB.

Since the effectiveness of phrases experiments is no better than the other DE runs and the resource requirements are comparatively large for phrase experiments, we did not experiment with phrases further.

**Effectiveness.** In the following discussion we treat any sub-collection as a full collection and neglect a change of 0.005 or less in the respective measurements. Across 13 collections, MAP was increased ten times through QE and decreased twice, whereas the DE technique $Q = D$ improved only five collections and degraded the results of one. These figures are six and two respectively for the term centric method. QE improved precision at 10 in nine instances and decreased it in three cases. Retrieval results for precision at 10 were increased three times and decreased in five instances, employing either DE technique. Using those terms for comparison, the summarisation technique performs the same as QE, with the exception of FT where it is a little worse the QE. Increases in effectiveness for DE methods are small compared to those of QE. Furthermore, improvements achieved by QE are mostly statistically significant whereas DE improvements are not.

**Efficiency.** The term centric approach slows down retrieval by 2% in most cases, whereas the document centric technique adds roughly 10%. These figures are the same on both machines as there is enough main memory on either machine for caching of inverted lists.

On $Lrg$, QE slows down retrieval by a factor of five to seven. Caching does not work well, since during query evaluation many lists have to be purged in order to make room for other lists and for documents that are retrieved from disk. This problem is exacerbated on $Ltl$ where the overhead increases from five to fifteen-fold.

The additional data needed for the technique involving summaries fits well into memory on $Lrg$, while leaving adequate room for inverted lists to be cached. This is why query times are increased only by around 50%. On $Ltl$, some of the in-memory summaries need to be swapped in and out of memory more often and the penalty is relatively high, leading to a decrease in query throughput to roughly half of that of the baseline.

**Robustness.** Figure 3 shows how many queries are degraded or improved in respect to the baseline and by how much. The baseline is constructed by running queries in their original form against the non-modified corpus. All lines more or less intersect the x-axis at the same point, which means that all methods examined in this paper exhibit roughly the same robustness for each collection.

| Coll. | Method | | MAP | P@10 | R-Pr. | | MAP | P@10 | R-Pr. | | MAP | P@10 | R-Pr. |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| WSJ2 | None | TREC 3 | 0.251 | 0.363 | 0.275 | | | | | | | | |
| | QE | | 0.325‡ | 0.388 | 0.324‡ | | | | | | | | |
| | $S = 40$ | | 0.286‡ | 0.380 | 0.287† | | | | | | | | |
| | $Q = D$ | | 0.265‡ | 0.361 | 0.280 | | | | | | | | |
| | $Q \in V$ | | 0.264 | 0.378 | 0.283 | | | | | | | | |
| | $Q = P$ | | 0.259 | 0.371 | 0.276 | | | | | | | | |
| | $Q = B$ | | 0.260 | 0.380 | 0.268† | | | | | | | | |
| AP | None | TREC 3 | 0.243 | 0.430 | 0.262 | | | | | | | | |
| | QE | | 0.327‡ | 0.468‡ | 0.333‡ | | | | | | | | |
| | $S = 40$ | | 0.290‡ | 0.454‡ | 0.301‡ | | | | | | | | |
| | $Q = D$ | | 0.251 | 0.416 | 0.286‡ | | | | | | | | |
| | $Q \in V$ | | 0.248 | 0.420 | 0.276‡ | | | | | | | | |
| NW | None | | | | | TREC 7 | 0.195 | 0.458 | 0.251 | TREC 8 | 0.222 | 0.438 | 0.262 |
| | QE | | | | | | 0.232‡ | 0.452 | 0.285‡ | | 0.250‡ | 0.464 | 0.289‡ |
| | $S = 40$ | | | | | | 0.208 | 0.438 | 0.263 | | 0.234 | 0.434 | 0.269 |
| | $Q = D$ | | | | | | 0.199 | 0.476 | 0.259 | | 0.213 | 0.444 | 0.263 |
| | $Q \in V$ | | | | | | 0.195† | 0.444 | 0.243 | | 0.220 | 0.434 | 0.261 |
| FBIS | None | TREC 6 | 0.223 | 0.260 | 0.232 | TREC 7 | 0.208 | 0.318 | 0.218 | TREC 8 | 0.269 | 0.319 | 0.281 |
| | QE | | 0.237‡ | 0.266 | 0.226 | | 0.222 | 0.292† | 0.243‡ | | 0.270 | 0.305 | 0.256 |
| | $S = 40$ | | 0.231 | 0.274 | 0.226 | | 0.217 | 0.308 | 0.224 | | 0.268 | 0.309 | 0.274 |
| | $Q = D$ | | 0.220‡ | 0.257 | 0.235 | | 0.205 | 0.300‡ | 0.218 | | 0.264‡ | 0.312 | 0.279 |
| | $Q \in V$ | | 0.233 | 0.260 | 0.237 | | 0.228 | 0.318 | 0.239‡ | | 0.278 | 0.321 | 0.284 |
| FT | None | TREC 6 | 0.214 | 0.250 | 0.244 | TREC 7 | 0.224 | 0.271 | 0.241 | TREC 8 | 0.290 | 0.331 | 0.298 |
| | QE | | 0.209 | 0.261 | 0.220 | | 0.233 | 0.287 | 0.234 | | 0.298 | 0.361† | 0.282 |
| | $S = 40$ | | 0.217 | 0.276‡ | 0.221 | | 0.216 | 0.269 | 0.229 | | 0.261 | 0.341 | 0.249 |
| | $Q = D$ | | 0.211† | 0.243 | 0.235 | | 0.229 | 0.277 | 0.242 | | 0.299 | 0.316 | 0.312 |
| | $Q \in V$ | | 0.206 | 0.237 | 0.229 | | 0.212 | 0.287‡ | 0.221 | | 0.295 | 0.325 | 0.304 |
| LA | None | TREC 6 | 0.198 | 0.231 | 0.232 | TREC 7 | 0.211 | 0.300 | 0.234 | TREC 8 | 0.233 | 0.260 | 0.238 |
| | QE | | 0.226‡ | 0.254‡ | 0.218 | | 0.251‡ | 0.316 | 0.269† | | 0.207 | 0.256 | 0.223 |
| | $S = 40$ | | 0.213‡ | 0.244‡ | 0.222 | | 0.240† | 0.306 | 0.263† | | 0.216 | 0.262 | 0.237 |
| | $Q = D$ | | 0.209 | 0.227 | 0.221 | | 0.225 | 0.304 | 0.242 | | 0.237 | 0.256 | 0.248 |
| | $Q \in V$ | | 0.216 | 0.237† | 0.237 | | 0.224 | 0.288 | 0.250 | | 0.235 | 0.256 | 0.242 |

Table 2: *Effectiveness of expansion techniques, averaged over 50 queries. The WSJ2 data was used for tuning. Shown are mean average precision (MAP), precision at 10 (P@10), and precision at the number of relevant documents (R-Pr.). Notation otherwise is the same as that used in Table 1. Results that are statistically significant different to the baseline at the 0.10 and 0.05 levels are marked with † and ‡ respectively.*

## 6 Analysis

An explanation for the relatively poor improvements of DE is that the topic of the expanded documents is changed too much from the original topic, analogous to query drift. This problem could be alleviated by adding a reduced weight to terms as they are added to documents. We leave this for future work.

A further explanation is that the lack of context during the expansion process is unhelpful; whereas, during conventional QE, several query terms set a particular context that determines the intersection of documents in the local set. Our experiments involving phrases try to address this problem. However, the generation method of phrases is most likely insufficient. Phrases are extracted from the collection itself – rather than from a suitable query log for example – and therefore no new context from outside the collection is found.

## 7 Conclusions

A series of experiments cannot prove that a family of methods is not viable. Establishing a positive result is straightforward; establishing a negative result involves demonstrating that all reasonable avenues of progress have been investigated and found wanting. Nonetheless, we believe we have shown that corpus-based DE is not promising. Other DE methods, based on extracting terms from external resources, have been found to give limited gains in some circumstances. However, while query-time costs are low, we were unable to use corpus-based DE to significantly improve effectiveness, and the index-time costs are considerable.

In contrast, our fresh investigation of QE showed that it was generally of benefit in the newswire collections used in our experiments, and that the evaluation costs can be much reduced while broadly maintaining the effectiveness gains. These results, we believe, should help focus future research in the area, by demonstrating that work on DE may not be warranted and by suggesting promising further directions for improving the efficiency and effectiveness of QE.

## Acknowledgements

We thank Nick Lester and William Webber. This research was conducted with the support of an APA, the State Government of Victoria, and an RMIT VRII grant.

## References

[1] B. Billerbeck, F. Scholer, H. E. Williams and J. Zobel. Query expansion using associated queries. In *Proc. Int. Conf. on Information and Knowledge Management*, pages 2–9, New Orleans, LA, November 2003. ACM Press, New York.

[2] B. Billerbeck and J. Zobel. Techniques for efficient query expansion. In A. Apostolico and M. Melucci (editors), *Proc. String Processing and Information Retrieval Symp.*, pages 30–42, Padova, Italy, September 2004. Springer-Verlag.

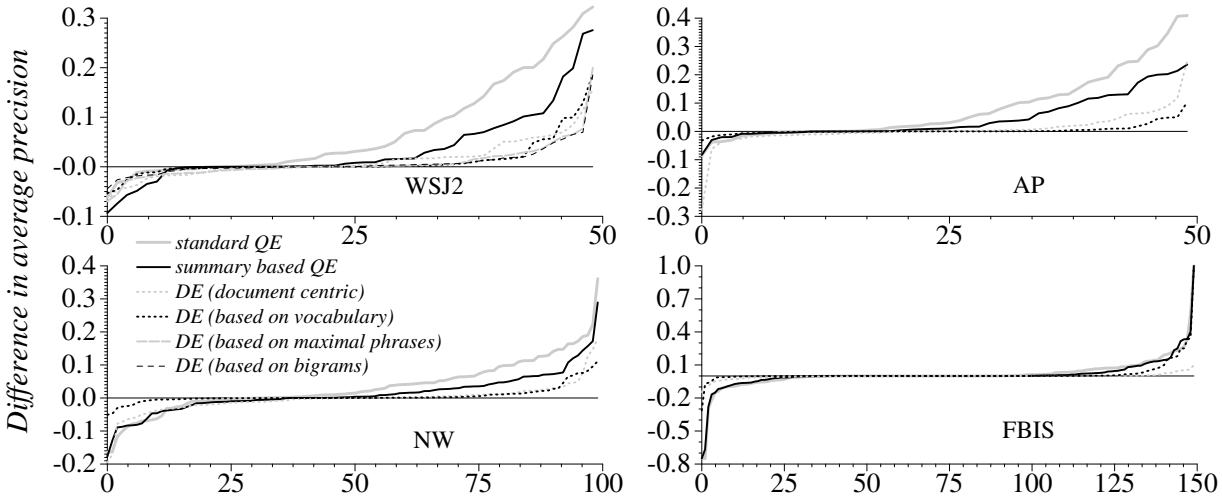


Figure 3: *The graphs show per-query-differences in average precision between each of the methods and the respective baselines. Each curve for each collection is sorted individually. Data for phrases and bigrams is only shown for WSJ2. The graphs for FT and LA are similar to that of FBIS and are not shown here.*


[3] W. B. Croft (editor). *Advances in Information Retrieval*. Kluwer Academic Publishers, Norwell, MA, 2000.

[4] S. Deerwester, S. T. Dumais, G. W. Furnas and T. Landauer. Indexing by latent semantic analysis. *Jour. of the American Society for Information Science*, Volume 41, Number 6, pages 391–407, 1990.

[5] E. Ide and G. Salton. Interactive search strategies and dynamic file organization in information retrieval. In G. Salton (editor), *The SMART Retrieval System – Experiments in Automatic Document Processing*, pages 373–393. Prentice-Hall, Englewood Cliffs, NJ, 1971.

[6] N. Lester and H. E. Williams. Topic tracking at RMIT University. In *Topic Detection and Tracking Workshop (TDT)*, Gaithersburg, MD, 2002. National Institute of Standards and Technology.

[7] G.-A. Levow and D. W. Oard. Signal boosting for translingual topic tracking: Document expansion and n-best translation. In *Topic Detection and Tracking: Event-Based Information Organization*, pages 175–195. Kluwer Academic Publishers, 2002.

[8] Y.-C. Li and H. M. Meng. Document expansion using a side collection for monolingual and cross-language spoken document retrieval. In *ISCA Workshop on Multilingual Spoken Document Retrieval*, pages 85–90, Hong Kong, 2003.

[9] L. A. F. Park and K. Ramamohanarao. Hybrid pre-query term expansion using latent semantic analysis. In *Proceedings of the fourth IEEE International Conference on Data Mining (ICDM'04)*, pages 178–185, Washington, DC, USA, 2004. IEEE Computer Society.

[10] S. E. Robertson and K. Sparck Jones. Relevance weighting of search terms. *Jour. of the American Society for Information Science*, Volume 27, Number 3, pages 129–146, 1976.

[11] S. E. Robertson and S. Walker. Okapi/Keenbow at TREC-8. In E. M. Voorhees and D. K. Harman (editors), *Proc. Text Retrieval Conf. (TREC)*, pages 151–161, Gaithersburg, MD, November 1999. National Institute of Standards and Technology Special Publication 500-246.

[12] J. J. Rocchio. Relevance feedback in information retrieval. In G. Salton (editor), *The SMART Retrieval System – Experiments in Automatic Document Processing*, pages 313–323. Prentice-Hall, Englewood Cliffs, NJ, 1971.

[13] I. Ruthven. Re-examining the potential effectiveness of interactive query expansion. In J. Callan, G. Cormack, C. Clarke, D. Hawking and A. Smeaton (editors), *Proc. ACM-SIGIR Int. Conf. on Research and Development in Information Retrieval*, pages 213–220, Toronto, Canada, July 2003. ACM Press, New York.

[14] F. Scholer, H. E. Williams and A. Turpin. Query association surrogates for web search. *Jour. of the American Society for Information Science and Technology*, Volume 55, Number 7, pages 637–650, 2004.

[15] A. Singhal and F. Pereira. Document expansion for speech retrieval. In F. Gey, M. Hearst and R. Tong (editors), *Proc. ACM-SIGIR Int. Conf. on Research and Development in Information Retrieval*, pages 34–41, Berkeley, CA, August 1999. ACM Press, New York.

[16] K. Sparck Jones, S. Walker and S. E. Robertson. A probabilistic model of information retrieval: Development and comparative experiments. Parts 1&2. *Information Processing & Management*, Volume 36, Number 6, pages 779–840, 2000.

[17] A. Spink, D. Wolfram, Major B. J. Jansen and T. Saracevic. From e-sex to e-commerce: Web search changes. *IEEE Computer*, Volume 35, Number 3, pages 107–109, March 2002.

[18] K. Taghva, J. Borsack, T. Nartker and A. Condit. The role of manually-assigned keywords in query expansion. *Information Processing & Management*, Volume 40, Number 3, pages 441–458, 2004.

[19] J. Xu and W. B. Croft. Improving the effectiveness of information retrieval with local context analysis. *ACM Transactions on Information Systems*, Volume 18, Number 1, pages 79–112, 2000.