Improving the Effectiveness of Information Retrieval with Local Context Analysis

JINXI XU
BBN Technologies
and
W. BRUCE CROFT
University of Massachusetts—Amherst

Techniques for automatic query expansion have been extensively studied in information retrieval research as a means of addressing the word mismatch between queries and documents. These techniques can be categorized as either global or local. While global techniques rely on analysis of a whole collection to discover word relationships, local techniques emphasize analysis of the top-ranked documents retrieved for a query. While local techniques have shown to be more effective than global techniques in general, existing local techniques are not robust and can seriously hurt retrieval when few of the retrieved documents are relevant. We propose a new technique, called *local context analysis*, which selects expansion terms based on cooccurrence with the query terms within the top-ranked documents. Experiments on a number of collections, both English and non-English, show that local context analysis offers more effective and consistent retrieval results.

Categories and Subject Descriptors: H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing—*Indexing methods*; *Thesauruses*; *Linguistic processing*; H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Query formulation*; *Search process*; *Relevance feedback*

General Terms: Experimentation, Performance

Additional Key Words and Phrases: Cooccurrence, document analysis, feedback, global techniques, information retrieval, local context analysis, local techniques

This work was completed when the first author was a Ph.D. student and then a postdoctoral research associate at the Center for Intelligent Information Retrieval in the Computer Science Department of University of Massachusetts—Amherst. It is partially based on chapters of the Ph.D. thesis of the first author. Some results also appeared in a SIGIR 96 paper by the authors.
Authors' addresses: J. Xu, BBN Technologies, 70 Fawcett Street, Cambridge, MA 02138; email: jxu@bbn.com; W. B. Croft, Computer Science Department, University of Massachusetts—Amherst, Amherst, MA 01003; email: croft@cs.umass.edu.

1. INTRODUCTION

A fundamental problem in information retrieval (IR) is word mismatch, which refers to the phenomenon that the users of IR systems often use different words to describe the concepts in their queries than the authors use to describe the same concepts in their documents. Word mismatch is a serious problem, as observed by Furnas et al. [1987] in a more general context. In their experiments, two people use the same term to describe an object less than 20% of the time. The problem is more severe for short casual queries than for long elaborate queries because as queries get longer, there is more chance of some important words cooccurring in the query and the relevant documents. Unfortunately, short queries are becoming increasingly common in retrieval applications, especially with the advent of the World Wide Web. Addressing the word mismatch problem has become an increasingly important research topic in IR.

In this article, we will discuss techniques that address the word mismatch problem through automatic query expansion. Automatic query expansion techniques have a significant advantage over manual techniques such as relevance feedback [Salton and Buckley 1990] and manual thesauri because they require no effort on the part of the user. Existing techniques for automatic query expansion can be categorized as either *global* or *local*. A global technique requires some corpuswide statistics that take a considerable amount of computer resources to compute, such as the cooccurrence data about all possible pairs of terms in a corpus (collection). The source of expansion terms is the whole corpus. A local technique processes a small number of top-ranked documents retrieved for a query to expand that query. A local technique may use some global statistics such as the document frequency of a term, but such statistics must be cheap to obtain. The source of expansion terms is the set of top-ranked documents.

One of the earliest global techniques is term clustering [Sparck Jones 1971], which groups words into clusters based on their cooccurrences and uses the clusters for query expansion. Other well-known global techniques include Latent Semantic Indexing [Deerwester et al. 1990], similarity thesauri [Qiu and Frei 1993], and Phrasefinder [Jing and Croft 1994]. Generally, global techniques did not show consistent positive retrieval results until better strategies for term selection were introduced in recent years. Global techniques typically need the cooccurrence information for every pair of terms. This is a computationally demanding task for large collections.

Local techniques expand a query based on the information in the set of top-ranked documents retrieved for the query [Attar and Fraenkel 1977; Buckley et al. 1995; Croft and Harper 1979]. The simplest local technique is local feedback [Buckley et al. 1995], which assumes the top-retrieved documents are relevant and uses standard relevance feedback procedures for query expansion. A similar and earlier technique was proposed in Croft and Harper [1979], where information from the top-ranked documents is used to reestimate the probabilities of query terms in the relevant set for a

query. The terms chosen by local feedback for query expansion are typically the most frequent terms (excluding stopwords) from the top-ranked documents. Recent TREC results show that local feedback can significantly improve retrieval effectiveness [Voorhees and Harman 1998]. Experiments have also shown that local feedback may not be a robust technique. It can seriously degrade retrieval performance if few of the top-ranked documents retrieved for the original query are relevant, because in this case the expansion terms will be selected mostly from nonrelevant documents.

In this article, we propose a new query expansion technique, called *local context analysis*. Although local context analysis is a local technique, it employs cooccurrence analysis, a primary tool for global techniques, for query expansion. The expansion features, called concepts, are extracted from the top-ranked documents. Concepts can be as simple as single terms and pairs of terms. More sophisticated concepts are nouns and noun phrases. Local context analysis ranks the concepts according to their cooccurrence within the top-ranked documents with the query terms and uses the top-ranked concepts for query expansion. Experimental results show that local context analysis produces more effective and more robust query expansion than existing techniques.

The remainder of the article is organized as follows: Sections 2 and 3 review existing techniques and point out the problems. Section 4 discusses local context analysis in detail. Section 5 outlines the experimental methodology and describes the test collections. Sections 6 to 14 present the experimental results. Section 15 discusses optimization and efficiency issues. Section 16 discusses other applications of local context analysis in IR. Section 17 draws conclusions and points out future work.

2. GLOBAL TECHNIQUES

2.1 Term Clustering

Global techniques for query expansion are typically based on the so-called association hypothesis, which states that words related in a corpus tend to cooccur in the documents of that corpus [van Rijsbergen 1979]. One of the earliest global techniques is term clustering, which groups related terms into clusters based on their cooccurrences in a corpus. The most representative work on term clustering was conducted by Sparck Jones in the late 60's and early 70's [Sparck Jones 1971]. She investigated a wide range of algorithms to form clusters and different methods to use them. Her major conclusion was that well-constructed term clusters can improve retrieval performance, but this was not supported by follow-up studies [Minker et al. 1972]. A serious problem with term clustering is that it cannot handle ambiguous terms. If a query term has several meanings, term clustering will add terms related to different meanings of the term and make the query even more ambiguous. In this case, it will lower retrieval effectiveness.

2.2 Dimensionality Reduction

Dimensionality reduction is related to term clustering. The most well-known technique of this type is Latent Semantic Indexing (LSI) [Deerwester et al. 1990; Furnas et al. 1988]. Other dimensionality reduction techniques were proposed in a number of studies, e.g., Caid et al. [1995] and Schütze and Pedersen [1994]. LSI decomposes a term into a vector in a low-dimensional space. This is achieved using a technique called singular value decomposition (SVD). It is hoped that related terms which are orthogonal in the high-dimensional space will have similar representations in the low-dimensional space, and as a result, retrieval based on the reduced representations will be more effective. Despite the potential claimed by its advocates, retrieval results using LSI so far have not shown to be conclusively better than those of standard vector space retrieval systems.

2.3 Phrasefinder

More recent global techniques address the word ambiguity problem by expanding a query as a whole. Two examples are Phrasefinder [Jing and Croft 1994] and Similarity Thesauri [Qiu and Frei 1993]. These techniques exploit the mutual disambiguation of the query terms by selecting expansion terms based on their cooccurrence with all query terms. Terms cooccurring with many query terms are preferred over terms cooccurring with few query terms.

We describe Phrasefinder in more detail, since it represents one of the most successful global techniques and is similar to the new query expansion technique proposed in this article. With Phrasefinder, a concept c (usually a noun phrase) is represented as a set of tuples $\{\langle t_1, a_1\rangle, \langle t_2, a_2\rangle, \ldots\}$, where t_i is a term cooccurring with c, while a_i is the number of cooccurrences between c and t_i. The set of tuples is called the pseudodocument of concept c. Given a query Q, the pseudodocuments of all concepts are ranked against Q as if they are real documents. The highest-ranked concepts are used for query expansion. As with any global technique, efficiency is a problem. The creation of the pseudodocuments requires the cooccurrence data for all possible concept-term pairs. The retrieval effectiveness of Phrasefinder is mixed. It is one of best global techniques, but judging from recent published results, it is not as effective as some local techniques [Xu and Croft 1996].

3. LOCAL TECHNIQUES

The idea of using the top-ranked documents for query expansion can be traced at least to a 1977 paper by Attar and Fraenkel [1977]. In that paper, term clustering was carried out by running traditional term-clustering algorithms on the top-ranked documents retrieved for a query. The term clusters were then used for query expansion. Attar and Fraenkel produced

positive improvement in retrieval effectiveness, but the test collection they used was too small to draw any definite conclusion.

A more recent local technique is local feedback (also known as pseudofeedback). Local feedback is motivated by relevance feedback [Salton and Buckley 1990], a well-known IR technique that modifies a query based on the relevance judgments of the retrieved documents. Relevance feedback typically adds common terms from the relevant documents to a query and reweights the expanded query based on term frequencies in the relevant documents and in the nonrelevant documents. Local feedback mimics relevance feedback by assuming the top-ranked documents to be relevant. The added terms are, therefore, common terms from the top-ranked documents. A similar and earlier technique proposed by Croft and Harper [1979] modifies the weights of the query terms but does not add new terms to a query. Local feedback has in recent years become a widely used query expansion technique. In the TREC conferences, for example, a large number of groups (Cornell University [Buckley et al. 1998], City University [Walker et al. 1998], Queens College at the City University of New York [Kwok et al. 1998], Australia National University [Hawking et al. 1998], etc.) have used some form of local feedback. In general, the TREC results show that local feedback is a simple yet effective query expansion technique. Its performance can be very erratic, however. It can seriously hurt retrieval if most of the top-ranked documents are not relevant.

There have been a number of efforts to improve local feedback. Mitra et al. [1998] attempted to improve the initial retrieval by reranking the retrieved documents using automatically constructed fuzzy Boolean filters. Lu et al. [1997] clustered the set of top-ranked documents and removed the singleton clusters in the hope of increasing the concentration of relevant documents in the reduced set. Buckley et al. [1998] also clustered the retrieved documents but used the clusters that best match the query for query expansion.

4. LOCAL CONTEXT ANALYSIS

4.1 Hypothesis

The most critical function of a local feedback algorithm is to separate terms in the top-ranked relevant documents from those in top-ranked nonrelevant documents. A common strategy uses the frequency statistics of terms in the top-ranked documents: The most frequent terms (except stopwords) in the top-ranked documents are used for query expansion. This strategy fails if there is a large cluster of nonrelevant documents in the top-ranked set. We will show that feature selection based on cooccurrence is a better strategy. We first make some hypotheses:

—*Hypothesis about the top-ranked set:* Top-ranked documents tend to form several clusters.

The conjecture that relevant documents tend to cluster was made by van Rijisbergen [1979]. This was recently supported by Hearst and Pedersen

[1996]. We conjecture that nonrelevant documents in the top-ranked set also tend to cluster. Unlike general nonrelevant documents, the top-ranked ones are retrieved based on similarity to a query. As a result, such nonrelevant documents may demonstrate some patterns. We can model a cluster of documents that are ranked highly for a query Q but are not relevant to it as the retrieval set for a different query Q'. Q and Q' share many terms but are about two completely different topics. For example, Q can be "DNA testing and crime," while Q' can be "DNA testing for treating cancer." If we search for the query "DNA testing and crime," we may retrieve many documents about "DNA testing for treating cancer."

The result of local feedback depends on whether the largest cluster is relevant. If it is, the frequent terms in the top-ranked set are also the frequent terms from the relevant cluster, and hence local feedback works. If it is not, the frequent terms are from a cluster of nonrelevant documents and hence local feedback fails. In one study, Lu et al. [1997] attempted to improve local feedback by removing singleton clusters from the top-ranked set. It is easy to see the problem with such a method: the removal of the singletons has no effect on whether the largest cluster is relevant.

—*Hypothesis about the relevant cluster in the top-ranked set:* The number of relevant documents that contain a query term is nonzero for (almost) every query term.

We think that the reason a user uses a term in a query is that he or she expects the term to be used in at least some relevant documents. Consider the hypothetical query "DNA testing and crime." Many relevant documents may not use the word "crime" (but use "murder" or "kill" instead). But if there are a reasonably large number of relevant documents, we expect a few of them will indeed use the word "crime." We should point out that we assume that there are a reasonably large number of relevant documents.

—*Hypothesis about a nonrelevant cluster in the top-ranked set:* For a cluster of nonrelevant documents, there are some query terms that do not occur in any of the documents in that cluster.

As we discussed in the first hypothesis, a nonrelevant cluster can be viewed as the retrieval result for a different query that shares many terms with the original query. Because the other query does not contain some terms in the original query, these terms will miss in every document in the cluster. For example, we expect that the query term "crime" never occurs in any document about "DNA testing for treating cancer."

The above hypotheses seem to suggest that we should cluster the top-ranked set and analyze the content of each cluster. We tried such a method, but it did not work well [Xu 1997]. The work by Buckley et al. [1998] basically found the same. A problem with document clustering is

that it can make mistakes. When this happens, it adds more noise to the query expansion process. Another problem is that we need to deal with the difficult issues such as how many clusters to create and what similarity threshold to use for cluster formation. Choosing the appropriate values for these parameters is crucial to the output of the clustering procedure.

Our approach uses clusters of documents but does not use a clustering algorithm. For each term t, we let $cluster(t)$ be the cluster of top-ranked documents that contain t. The above hypotheses imply that if t is a common term from the top-ranked relevant documents, then $cluster(t)$ will tend to contain all query terms. That is, t will tend to cooccur with all query terms within the top-ranked set. In summary, our general hypothesis is as follows:

HYPOTHESIS. A common term from the top-ranked relevant documents will tend to cooccur with all query terms within the top-ranked documents.

We will use the above hypothesis as a guideline when we define a metric for selecting expansion terms in the next section.

4.2 Metric for Concept Selection

The expansion features selected by local context analysis are called concepts. We now discuss the metric used by local context analysis for concept selection. We assume that (1) the query to be expanded is Q, that (2) the query terms in Q are $w_1, w_2 \ldots w_m$, and (3) that the collection being searched is C. We denote the set of top-ranked documents as $S = \{d_1, d_2, \ldots d_n\}$. (The retrieval system used in this article is INQUERY [Broglio et al. 1994].) Based on our hypothesis, we should use concepts that cooccur with all query terms for query expansion. In practice, however, a query term may not be used in any relevant document. Our approach is to prefer concepts cooccurring with more query terms over those cooccurring with fewer query terms for query expansion. Specifically, we will derive a function $f(c, Q)$ which measures how good a concept c is for expanding query Q based on c's cooccurrence with w_i's in S. All concepts are ranked by f, and the best concepts are added to Q.

—*Cooccurrence metric:* We hypothesized that good expansion concepts tend to cooccur with all query terms in the top-ranked documents. But we must take random cooccurrences into account: a concept c could just cooccur with a query term w_i in the top-ranked documents by chance. The higher the concept c's frequency in the whole collection, the more likely it is that it cooccurs with w_i by chance. The larger the number of cooccurrences, the less likely that c cooccurs with w_i by chance. Let N be the number of documents in C, N_c the number of documents that contain c, and $co(c, w_i)$ the number of cooccurrences between c and w_i in S. We proposed the following metric to measure the degree of cooccurrence of c with w_i:

$$co\_degree(c, w_i) = \log_{10}(co(c, w_i) + 1)idf(c)/\log_{10}(n)$$

$$co(c, w_i) = \sum_{d\ in\ S} tf(c, d)\ tf(w_i, d)$$

$$idf(c) = min(1.0, \log_{10}(N/N_c)/5.0)$$

where $tf(c, d)$ and $tf(w_i, d)$ are the frequencies of c and w_i in document d, respectively. We view the metric as the likelihood that concept c and query term w_i cooccur nonrandomly in the top-ranked set S, though it is not a probability in a strict sense. The metric takes into account the frequency of c in the whole collection ($idf(c)$) and the number of cooccurrences between c and w_i in the top-ranked set ($co(c, w_i)$). The logarithm function is used to dampen the raw numbers of occurrences and cooccurrences. The metric is normalized over n (the number of documents in the top-ranked set). If each term occurs at most once in a document, n is an upper bound for the raw number of cooccurrences.

We should point out that the above cooccurrence metric is different from well-known metrics such as EMIM (expected mutual information measure) [Church and Hanks 1989; van Rijsbergen 1979], cosine [Salton 1989], and so forth. One reason we choose not to use the available metrics is that they are designed to measure corpuswide cooccurrence, and it is not clear how to adapt them to measure cooccurrence in the top-ranked documents. The other reason is that we want to explicitly bias against high-frequency concepts, but available metrics cannot do that.

We should also point out that our definition of idf is somewhat different from the standard definition $idf(c) = \log_{10}(N/N_c)$ used by other researchers. The problem with the latter definition is that mathematically it has no upper limit when N approaches infinity. Our formula sets an upper limit on $idf(c)$. Any concept which occurs in 1/100,000 of the documents or less frequently will have idf 1.0. A similar idf function was used by Kwok [1996]. Greiff [1998] explained why setting an upper bound on idf is desirable.

—*Combining the degrees of cooccurrence with all query terms:* If $co\_degree(c, w_i)$ measures the likelihood that concept c cooccurs with query term w_i nonrandomly in the top-ranked set, we now need to estimate the likelihood that c cooccurs nonrandomly with all w_i's in the top-ranked set. Assuming c's cooccurrences with different query terms are independent, the natural estimate is to multiply the m $co\_degree(c, w_i)$ values. A problem with such an estimate is that if one of the m numbers is 0, the product is 0 no matter what the other $m - 1$ numbers are. A more desirable function should produce a nonzero value based on the other $m - 1$ numbers. For this purpose, we add a small constant δ to each degree of cooccurrence. The function for combining the m numbers is therefore

$$g(c, Q) = \prod_{w_i \text{ in } Q} (\delta + co\_degree(c, w_i)).$$

The use of δ in g is a simple smoothing technique. Smoothing is widely used in various statistical models (including IR models) which deal with a limited amount of data. For example, INQUERY uses a default belief (typically 0.4) to prevent zero values from its #AND operator when one operand is zero.

From another perspective, because of δ, g achieves a "relaxed" interpretation of the Boolean statement that good concepts cooccur with all query terms. To simplify, we assume Q has only two terms w_1 and w_2. We can rewrite $g(c, Q)$ as follows:

$$\begin{aligned} g(c, Q) = {} & co\_degree(c, w_1) co\_degree(c, w_2) \\ & + \delta(co\_degree(c, w_1) + co\_degree(c, w_2)) \\ & + \delta^2 \end{aligned}$$

g consists of three parts. The first part, $co\_degree(c, w_1) co\_degree(c, w_2)$, emphasizes cooccurrence with all query terms. The second part, $\delta(co\_degree(c, w_1) + co\_degree(c, w_2))$, emphasizes cooccurrence with individual query terms. The third part, δ^2, has no effect on the ranking of the concepts. The relative weights of the first and second parts are controlled by the δ-value. With a small δ, concepts cooccurring with all query terms are ranked higher. With a large δ, concepts having significant cooccurrences with individual query terms are ranked higher. In a sense, g is similar to the *AND* operator in the p_norm model [Fox 1983]. The purpose of δ in g is the same as that of p in the p_norm model.

—*Differentiating rare and common query terms:* Obviously, not all query terms are equally important. While deciding the importance of a query term is a hard problem in general, it is well-known that rare terms are usually more important than frequent ones. This is the reason behind the $tf \times idf$ formula used by most IR systems. Taking into account the idf of the query terms, we get the following function:

$$f(c, Q) = \prod_{w_i \text{ in } Q} (\delta + co\_degree(c, w_i))^{idf(w_i)}$$

Function f is used by local context analysis for ranking concepts. Because the idf values of the query terms are used as exponents in the formula, cooccurrence with a rare query term will have a bigger impact on $f(c, Q)$ than cooccurrence with a frequent query term. Note that multiplying $idf(w_i)$ with $co\_degree(c, w_i)$ does not work because, given a query, it only scales g by a constant factor.

```
#WSUM(1
      1 #WSUM (1 1 status 1 nuclear 1 proliferation 1 treaties
                 1 violations 1 monitoring)
      2 #WSUM (1
               1        #PHRASE(nuclear non proliferation treaty)
               0.987143 treaty
               0.974286 weapon
               0.961429 pakistan
               0.948571 missile
               0.935714 iraq
               0.922857 proliferation
               0.91     #PHRASE(non proliferation treaty)
               0.897143 #PHRASE(international atomic energy agency)
               0.884286 india
               0.871429 warhead
               0.858571 uranium
               0.845714 disarmament
               0.832857 china
               0.82     #PHRASE(chemical weapon)
               0.807143 spread
               ...
               ))
```

Fig. 1. Query expansion by local context analysis for TREC topic 202 "Status of nuclear proliferation treaties, violations and monitoring." #PHRASE is an INQUERY operator to construct phrases.

The concept selection metric f was heuristically derived based on our hypothesis that a good expansion term will tend to cooccur with all query terms. Although the metric largely reflects our hypothesis and works well in practice (as we will show later), a theoretically motivated metric would be more desirable. We have made substantial progress in developing such a metric based on language models. Language modeling is a powerful tool for speech recognition and more recently has been successfully applied to IR problems [Ponte 1998].

In summary, local context analysis takes these steps to expand a query Q on a collection C:

(1) Perform an initial retrieval on C to get the top-ranked set S for Q.

(2) Rank the concepts in the top-ranked set using the formula $f(c, Q)$.

(3) Add the best k concepts to Q.

Figure 1 shows an example query expanded by local context analysis.

4.3 Passages and Concepts

Local context analysis can use either the top-ranked documents or the top-ranked passages for query expansion. The default is to use the top-ranked passages in order to provide faster query expansion. A long document sometimes contains only one or two paragraphs that are related to a query. Using passages avoids processing the unnecessary parts of the

document and saves time. Passages are created by breaking documents into fixed-length text windows, typically 300 words long. Experiments show that using documents and passages works equally well.

In the default mode, local context analysis selects nouns and noun phrases as the expansion concepts. For English text, nouns and noun phrases are recognized by a parts of speech tagger, *Jtag* [Xu et al. 1994]. Local context analysis can also select ordinary terms. This is useful when parts of speech software is unavailable for a language (e.g., Chinese). The main reason we use noun phrases as the default is that they are more suitable for user interaction. It appears that users are more accustomed to expressing their information needs in noun phrases than in other types of terms. Croft et al. [1995] compiled a list of the most common queries people used to search a collection of government records. An overwhelming majority of the queries are a noun compound. Using noun phrases is, therefore, an advantage when the user can read and modify the expanded query if necessary. Although user interaction is not a concern in this study, we considered that when we implemented local context analysis. From the viewpoint of retrieval effectiveness, however, using noun phrases is not an advantage. Query expansion using ordinary terms produces equally good results.

In summary, the major difference of local context analysis from existing local techniques lies not in using passages and concept analysis, but in a new metric for selecting expansion terms.

5. EXPERIMENTAL METHODOLOGY

Table I lists the test collections used in the experiments in this article. The TREC3 queries consist of the title, description, and narrative fields of the TREC topics while the TREC4 and TREC5 queries only use the description field. These test collections have quite different characteristics. The TREC3 queries are very long, averaging 34.5 words per query. The TREC5 queries are much shorter, averaging only 7.1 words per query. The WEST documents are very long, more than 10 times longer than TREC5-SPANISH documents on average. The TREC3 queries have far more relevant documents than the WEST queries. The WEST collection is homogeneous in that its documents have similar types and similar content. Other collections are more heterogeneous and contain documents of different types, different content, different lengths, and different sources. The collections are in three languages: English, Spanish, and Chinese. It is well known that many IR techniques are sensitive to factors such as query length [Wilkinson et al. 1996], document length [Singhal et al. 1996], collection size, and so forth. The purpose of using a wide variety of collections is to ensure the generality of the conclusions we reach in this article.

We will compare the performance of local context analysis not only with the performance of the original (unexpanded) queries, but also with the performance of local feedback and that of Phrasefinder. The main evaluation metric is interpolated 11-point average precision. Statistical t-test

Table I. Statistics about Test Collections. Stopwords are not included. Each Chinese character is counted as a word.

Collection	Query Count	Size (GB)	Document Count	Words per Query	Words per Document	Relevant Documents per Query
WEST	34	0.26	11,953	9.6	1967	28.9
TREC3	50	2.2	741,856	34.5	260	196
TREC4	49	2.0	567,529	7.5	299	133
TREC5	50	2.2	524,929	7.1	333	110
TREC5- SPANISH	25	0.34	172,952	8.2	156	100
TREC5- CHINESE	19	0.17	164,779	21	411	73.6

[Hull 1993] and query-by-query analysis are also employed. To decide whether the improvement by method A over method B is significant, the t-test calculates a p-value based on the performance data of A and B. The smaller the p-value, the more significant is the improvement. If the p-value is small enough (p-value < 0.05), we conclude that the improvement is statistically significant.

For local context analysis, the default is to use 70 concepts from 100 top-ranked passages for query expansion. The default concepts are nouns and noun phrases. The default passage size is 300 words. The default δ-value is 0.1. Concepts are added to a query according to the following formulas:

$$Q_{new} = \#\text{WSUM}\ (1.0\ 1.0\ Q_{old}\ wt\ Q')$$

$$Q' = \#\text{WSUM}\ (1.0\ wt_1\ c_1\ wt_2\ c_2 ... wt_{70}\ c_{70})$$

$$wt_i = 1.0 - 0.9i/70$$

where c_i is the ith ranked concept. Concepts are weighted in proportion to their ranks so that a higher-ranked concept is weighted more heavily than a lower-ranked concept. We call Q' the auxiliary query. The default value for wt is 2.0. #WSUM is an INQUERY operator to combine evidence from different parts of a query. Specifically, it computes a weighted average of its operands. We found that the above parameter settings work well in our experiments. We should stress, however, that the effectiveness of local context analysis is very stable with relation to the parameters settings: changing the parameters does not substantially affect retrieval performance.

Experiments with local feedback and Phrasefinder are carried out using established parameter settings for the two techniques. The local feedback experiments are based on the procedure used by the Cornell group in the TREC4 conference [Buckley et al. 1996]. It represents one of the most successful local feedback techniques used in the TREC conferences. The most frequent 50 single words and 10 phrases (pairs of adjacent nonstop-

Table II. A Comparison of Baseline, Phrasefinder, Local Feedback, and Local Context Analysis on TREC3. Ten documents for local feedback (lf-10doc). One hundred passages for local context analysis (lca-100p).

	Precision (% change)—50 Queries			
Recall	Base	Phrasefinder	lf-10doc	lca-100p
0	82.2	79.4 (−3.3)	82.5 (+0.4)	87.0 (+5.9)
10	57.3	60.1 (+4.8)	64.9 (+13.3)	65.5 (+14.3)
20	46.2	50.4 (+9.1)	56.1 (+21.5)	57.2 (+23.8)
30	39.1	43.3 (+10.7)	48.3 (+ 23.5)	48.4 (+23.8)
40	32.7	36.9 (+12.8)	41.6 (+26.9)	42.7 (+30.4)
50	27.5	31.8 (+15.9)	36.8 (+34.1)	37.9 (+38.0)
60	22.6	26.1 (+15.1)	30.9 (+36.7)	31.5 (+39.3)
70	18.0	20.6 (+14.0)	25.2 (+40.0)	25.6 (+42.1)
80	13.3	15.8 (+18.6)	19.4 (+45.7)	19.4 (+45.7)
90	7.9	9.4 (+18.7)	11.5 (+44.3)	11.7 (+47.3)
100	0.5	0.8 (+60.9)	1.2 (+143.5)	1.4 (+177.0)
Average	31.6	34.1 (+7.8)	38.0 (+20.5)	38.9 (+23.3)

words) from the top-ranked documents are added to a query. The words and phrases in the expanded query are then reweighted using the Rocchio weighting method [Rocchio 1971] with $\alpha : \beta : \gamma = 1 : 1 : 0$. We also applied local feedback on the top-ranked passages in order to compare with local context analysis. The Phrasefinder experiments are based on the method described in the UMass TREC3 report [Broglio et al. 1995]: 30 concepts are added to a query and are weighted in proportion to their rank position. Phrasefinder as a query expansion technique was documented in [Jing and Croft 1994]. Concepts containing only terms in the original query are weighted more heavily than those containing terms not in the original query.

6. EXPERIMENTAL RESULTS

We now present the experimental results of three query expansion techniques: Phrasefinder, local feedback,[1] and local context analysis. The experiments were carried out on TREC3, TREC4, TREC5, and WEST (Tables II–V). In the experiments, 10 documents were used per query for local feedback and 100 passages per query for local context analysis. Other parameters took their default value. A Phrasefinder result for TREC5 is not available. The original (unexpanded) queries were used as the baseline in the experiments.

[1]Mitra et al. [1998] reported better results using local feedback. The performance difference may be caused by differences in retrieval functions, indexing, and query processing. Another potential cause for the difference is that the local feedback techniques are somewhat different. Ours is based on the Cornell TREC4 report [Buckley et al. 1996] and does not assume any nonrelevant documents for a query while theirs is based on more recent Cornell TREC work [Buckley et al. 1998] and assumes some nonrelevant documents for a query.

Table III. A Comparison of Baseline, Phrasefinder, Local Feedback, and Local Context Analysis on TREC4. Ten documents for local feedback (lf-10doc). One hundred passages for local context analysis (lca-100p).

	Precision (% change)—49 Queries			
Recall	Base	Phrasefinder	lf-10doc	lca-100p
0	71.0	68.6 (−3.3)	68.4 (−3.6)	73.2 (+3.2)
10	49.3	48.6 (−1.6)	52.8 (+7.0)	57.1 (+15.7)
20	40.4	40.0 (−1.0)	43.2 (+7.0)	46.8 (+16.0)
30	33.3	33.9 (+1.8)	36.0 (+8.0)	39.9 (+19.8)
40	27.3	28.0 (+2.5)	29.8 (+9.2)	35.3 (+29.1)
50	21.6	23.9 (+10.3)	24.5 (+13.2)	29.9 (+38.4)
60	14.8	18.8 (+27.1)	19.7 (+33.4)	23.6 (+59.8)
70	9.5	11.8 (+24.7)	14.8 (+56.9)	17.9 (+89.1)
80	6.2	8.1 (+31.0)	10.8 (+74.7)	11.8 (+91.0)
90	3.1	4.2 (+33.6)	6.4 (+104.6)	5.7 (+80.2)
100	0.4	0.6 (+24.0)	0.9 (+93.3)	0.8 (+88.2)
Average	25.2	26.0 (+3.4)	27.9 (+11.0)	31.1 (+23.5)

Table IV. Retrieval performance on TREC5. Ten documents are used for local feedback (lf-10doc). One hundred passages are used for local context analysis (lca-100p).

	Precision (% change)—50 Queries		
Recall	Base	lf-10doc	lca-100p
0	64.1	48.5 (−24.4)	53.7 (−16.2)
10	37.5	36.8 (−1.9)	34.4 (−8.4)
20	29.1	31.0 (+6.5)	30.9 (+6.3)
30	24.1	26.5 (+9.9)	26.4 (+9.3)
40	21.3	22.6 (+6.2)	23.5 (+10.5)
50	17.9	19.3 (+7.8)	20.7 (+15.8)
60	12.6	15.6 (+24.2)	17.1 (+36.2)
70	10.1	12.9 (+27.6)	12.7 (+25.2)
80	7.2	9.4 (+31.5)	8.7 (+21.6)
90	4.8	6.6 (+36.3)	6.2 (+28.2)
100	2.7	2.7 (−2.3)	2.4 (−13.5)
Average	21.0	21.1 (+0.2)	21.5 (+2.3)

On TREC3, local context analysis is 23.3% better then the baseline, which is statistically significant (p-value = 0.000005). Local context analysis is also better than Phrasefinder by 14% and local feedback by 2.4%. The improvement by local context analysis over Phrasefinder is statistically significant (p-value = 0.0003), but the improvement over local feedback is not. Query expansion is generally regarded as a recall device, but the results show that precision at low recall points is improved as well. The reason for the improved precision is that some relevant documents which are ranked low by the original queries are pushed to the top of the ranked output because they contain many expansion concepts. The results show that query expansion potentially can be a precision device.

Table V. A Comparison of Baseline, Phrasefinder, Local Feedback, and Local Context Analysis on WEST. Ten documents are used for local feedback (lf-10doc). Rocchio β parameter is reduced by 50% in lf-10doc-dw. One hundred passages are used for local context analysis (lca-100p). Expansion concepts are downweighted by 50% in lca-100p-dw.

	Precision (% change)—34 queries					
Recall	Base	Phrasefinder	lf-10doc	lf-10doc-dw	lca-100p	lca-100p-dw
0	88.0	83.9 (−4.7)	80.7 (−8.3)	81.9 (−7.0)	91.9 (+4.4)	92.1 (+4.7)
10	80.0	74.5 (−6.9)	76.0 (−5.0)	76.9 (−4.0)	85.7 (+7.1)	84.3 (+5.4)
20	77.5	67.2 (−13.3)	70.5 (−8.9)	71.4 (−7.8)	76.3 (−1.5)	78.5 (+1.3)
30	74.1	64.3 (−13.2)	66.8 (−9.8)	68.2 (−7.9)	71.1 (−4.0)	73.9 (−0.1)
40	62.9	54.6 (−13.2)	59.5 (−5.4)	60.8 (−3.3)	61.3 (−2.6)	61.8 (−1.7)
50	57.5	49.5 (−14.0)	54.2 (−5.8)	56.8 (−1.2)	55.2 (−3.9)	56.8 (−1.2)
60	49.7	44.6 (−10.1)	45.5 (−8.3)	50.1 (+0.8)	49.2 (−0.8)	50.7 (+2.2)
70	41.5	37.4 (−9.9)	38.7 (−6.8)	42.1 (+1.3)	41.1 (−1.1)	44.2 (+6.4)
80	32.7	30.0 (−8.2)	29.2 (−10.8)	33.1 (+1.1)	33.6 (+2.6)	36.4 (+11.2)
90	19.3	18.4 (−4.6)	18.1 (−6.1)	21.8 (+13.0)	21.5 (+11.7)	22.6 (+17.1)
100	8.6	8.7 (+0.3)	8.2 (−4.8)	9.3 (+7.8)	9.6 (+10.9)	10.0 (+15.3)
Average	53.8	48.5 (−9.9)	49.8 (−7.5)	52.0 (−3.3)	54.2 (+0.8)	55.6 (+3.3)

On TREC4, local context analysis is significantly better than the baseline (23.5%, p-value = 0.00000006), Phrasefinder (19.6%, p-value = 0.00001), and local feedback (11.5%, p-value = 0.001).

On TREC5, local context analysis is 2.3% better than the baseline and 2.1% better than local feedback. The improvements are not statistically significant. We think that the TREC5 result is less significant mainly because of the peculiarities of the TREC5 query set. A number of queries are ill-formed from the retrieval point of view in that they use terms which are poor or even negative indicators of relevance. Examples include "existence of human life 10,000 years ago" (which is highly ambiguous) and "Identify social programs for poor people in countries other than the U.S." ("U.S." is negative evidence of relevance). For such queries, local context analysis is likely to choose bad concepts for query expansion by requiring them to cooccur with the bad query terms. Furthermore, three of the TREC5 queries have only one or two relevant documents in the whole collection. (If we remove these queries, the improvement will be 10.1%.) Local context analysis assumes that there are a reasonable number of relevant documents. Since the assumption is violated, local context analysis fails to improve them. Please note local feedback fails to improve them too.

On WEST, local context analysis is 0.8% better than the baseline, which is not statistically important. But it is significantly better than Phrasefinder (11.8%, p-value = 0.00003) and local feedback (8.8%, p-value = 0.002). The reason for the small improvement is due to the high-quality of the original queries. A comparison of the WEST and TREC queries shows that the WEST queries are more precise descriptions of the information needs. This is also shown by the very good performance of the original queries. Word mismatch is a less serious problem, and therefore query expansion is less useful. But even with such high-quality queries, local

context analysis manages to produce a small improvement. The improvement at low recall points is more noticeable. In contrast, Phrasefinder and local feedback seriously degrade the WEST queries. Since the original queries are very good, we conjectured that better retrieval is possible if we downweighted the expansion concepts. This is supported by retrieval results: when the weights of expansion concepts are reduced by 50%, local context analysis produces a 3.3% improvement over the baseline. When we reduce the β parameter of the Rocchio weighting formula by 50%, the result of local feedback is also improved, but it is still 3.3% worse than the baseline. In the remainder of the article, we will always downweight the expansion features by 50% for local context analysis and local feedback on WEST.

As we mentioned before, a problem with local feedback is its inconsistency. It can improve some queries and seriously hurt others. A query-by-query analysis on TREC4 shows that local context analysis is better than local feedback in this aspect. Although both techniques significantly improve retrieval effectiveness on TREC4, local context analysis improves more queries and hurts fewer than local feedback. Of 49 TREC4 queries, local feedback hurts 21 and improves 28, while local context analysis hurts 11 and improves 38. Of the queries hurt by local feedback, 5 queries have more than 5% loss in average precision. In the worst case, the average precision of one query is reduced from 24.8% to 4.3%. Of those hurt by local context analysis, only one has more than 5% loss in average precision. Local feedback also tends to hurt queries with poor performance. Of 9 queries with baseline average precision less than 5%, local feedback hurts 8 and improves 1. In contrast, local context analysis hurts 4 and improves 5.

Overall, experimental results show that local context analysis is a more effective and more consistent query expansion technique than local feedback and Phrasefinder. Of the three techniques, Phrasefinder is the least effective. The results show, that, while analysis of the top-ranked documents/passages is more effective than analysis of a whole corpus, the cooccurrence analysis used by global techniques can make local techniques more effective and more consistent.

7. THE IMPACT OF USING PASSAGES AND NOUN PHRASES

As we mentioned before, the benefits of using passages and noun phrases are faster query expansion and better user interaction, but this has little impact on retrieval effectiveness. First, we examine the impact of using noun phrases. When we let local context analysis extract single words and pairs of words, the retrieval performance on TREC4 is only 0.2% worse than using nouns and noun phrases (Table VI).

Second, we examine the impact of using passages. Table VII shows the retrieval results (average precision and improvement over the baseline) on TREC4 when both documents and 300–word passages are used for local context analysis. Several different numbers of passages and documents were used in order to make a thorough comparison. Using passages has no

Table VI. TREC4 Results: Comparing Noun Phrases with Words and Pairs of Words on the Performance of Local Context Analysis. One hundred passages are used.

	Precision (% change)—49 queries	
Recall	Noun Phrases	Terms and Pairs
0	73.2	75.8 (+3.5)
10	57.1	56.9 (−0.2)
20	46.8	46.9 (+0.1)
30	39.9	39.3 (−1.4)
40	35.3	33.5 (−4.9)
50	29.9	28.2 (−5.9)
60	23.6	22.2 (−5.9)
70	17.9	18.0 (+0.5)
80	11.8	12.9 (+9.9)
90	5.7	6.9 (+21.1)
100	0.8	0.8 (−3.6)
Average	31.1	31.0 (−0.2)

Table VII. TREC4 Results: Using 300-Word Passages versus Using Whole Documents on the Performance of Local Context Analysis

	Number of Passages/Documents Used					
	10	20	30	50	70	100
300-Word Passages	29.5 +17	29.9 +18.6	30.2 +19.8	30.4 +20.6	30.5 +21.	31.1 +23.5
Whole Documents	30.2 +19.9	30.9 +22.6	30.8 +22.5	30.8 +22.4	31.1 +23.4	30.7 +21.9

advantage over using whole documents: the best run of each method has almost identical performance (31.1%). However, it requires fewer documents than passages to achieve the best performance (70 documents versus 100 passages). This is understandable because documents are longer than passages.

Experiments show that using passages also has little impact on the performance of local feedback. In the experiments, we treated the top-ranked passages as if they were documents and applied local feedback. The best run of using passages is close to that of using whole documents (Table VIII).

To summarize, we conclude that the performance improvement by local context analysis over local feedback we observed in the previous section is due to a better metric for selecting expansion terms.

8. LOCAL CONTEXT ANALYSIS VERSUS LOCAL FEEDBACK

8.1 Varying the Number of Passages/Documents Used

One of the parameters in local context analysis and local feedback is how many top-ranked passages/documents to use for a query. So far we have not found a satisfactory method to automatically determine the optimal num-

Table VIII. TREC4 Results: Using 300-Word Passages versus Using Whole Documents on the Performance of Local Feedback

	Number of Passages/Documents Used					
	5	10	20	30	50	100
300-Word Passages	28.6 +13.6	28.7 +13.9	28.4 +12.7	27.6 +9.6	26.9 +6.8	25.4 +0.8
Whole Documents	28.7 +14.0	27.9 +11.0	26.9 +6.8	27.2 +8.2	26.7 +6.2	26.1 +3.5

Table IX. The Impact of the Number of Passages Used per Query on the Performance of Local Context Analysis

	Number of Passages							
Collection	10	20	30	50	100	200	300	500
TREC3	36.6 +16	37.5 +18.9	38.7 +22.6	38.9 +23.2	38.9 +23.3	39.3 +24.4	39.1 +23.7	38.3 +21.3
TREC4	29.5 +17	29.9 +18.6	30.2 +19.8	30.4 +20.6	31.1 +23.5	31.0 +23.0	30.7 +21.8	29.9 +18.6
TREC5	23.0 +9.2	23.0 +9.2	22.5 +6.8	21.1 +0.3	21.5 +2.3	21.1 +0.1	20.8 −1.0	20.9 −0.9
WEST	55.9 +3.8	56.5 +5.0	55.6 +3.4	55.8 +3.7	55.6 +3.3	54.6 +1.6	54.4 +1.2	53.6 −0.4

Table X. The Impact of the Number of Documents Used per Query on the Performance of Local Feedback

	Number of Documents Used					
Collection	5	10	20	30	50	100
TREC3	36.6 +16.0	38.0 +20.5	37.6 +19.1	37.7 +19.4	37.7 +19.3	36.6 +15.8
TREC4	28.7 +14.0	27.9 +11.0	26.9 +6.8	27.2 +8.2	26.7 +6.2	26.1 +3.5
TREC5	21.1 +0.5	21.1 +0.2	19.3 −8.2	19.4 −7.9	19.4 −7.6	17.8 −15.2
WEST	52.6 −2.2	52.0 −3.3	48.7 −9.5	47.5 −11.6	44.5 −17.2	40.0 −25.7

ber of passages/documents on a query-by-query basis. Until we find a solution, we would prefer that a technique does not rely too heavily on the particular value chosen for the parameter. In other words, a desirable technique should work well for a wide range of choices.

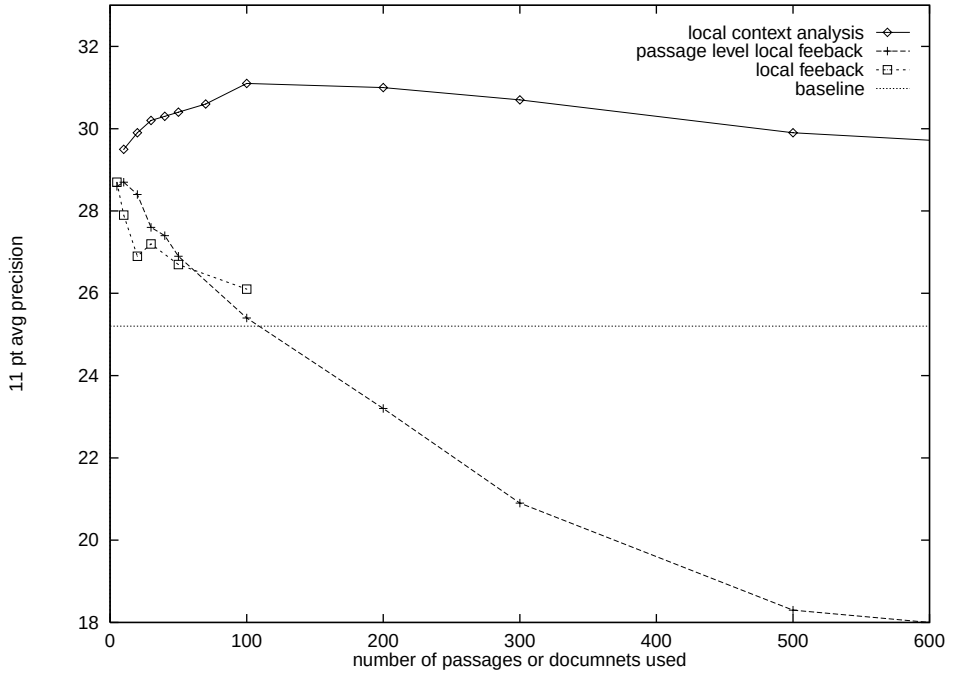

Fig. 2. Performance curves of local context analysis and local feedback on TREC4.

First we vary the number of top-ranked passages used per query and check the impact on the performance of local context analysis. The results are shown in Table IX. For each collection and each choice of the number of passages, the table shows the average precision and the improvement over the baseline. The results show that local context analysis works for a wide range of choices except for TREC5 (as we discussed before, the TREC5 query set should be considered an exception). For TREC3 and TREC4, any choice between 30 to 300 works well.

Table X shows the impact of the number of top-ranked documents used per query on the performance of local feedback. Local feedback depends heavily on the number of documents used per query, except for TREC3.

In general, the results show that local context analysis is less sensitive to the number of passages/documents used than local feedback. The difference between the two techniques in this aspect is more clearly shown in Figure 2. From this figure we can see another difference between the two techniques: their performances peak at quite different numbers of passages/documents. The performance of local feedback peaks when relatively few documents are used per query, while the performance of local context analysis peaks when significantly more passages are used. On TREC4, increasing the number of documents/passages from 10 to 100 hurts one technique but improves the other. The same is observed even if the top-ranked passages are used for local feedback. The difference can be explained by the assumptions made by the two techniques about the top-ranked set. The assumption made by local feedback (i.e., the relevant

cluster is the largest) is less likely to hold with a larger top-ranked set. As we enlarge the set, the percentage of nonrelevant documents will increase. The chance of retrieving large clusters of nonrelevant documents will also increase. On the other hand, the assumption made by local context analysis (i.e., a reasonable number of top-ranked documents are relevant) is more likely to hold with a larger top-ranked set, because increasing the top-ranked set will also increase the number of relevant documents. That is why a large top-ranked set hurts one technique but helps the other.

8.2 Dependence on the Quality of the Top-Ranked Set

Although both local context analysis and local feedback assume that some top-ranked documents/passages are relevant, experimental results on TREC4 show that local context analysis is less heavily dependent on the assumption than local feedback. We let both techniques use the top-ranked 100 passages for query expansion. We define a relevant passage as a passage from a relevant document. The average number of relevant passages in the top-ranked set is 26 per query. In the discussion, we are only interested in queries which are improved or degraded by a technique, with performance change of 1% or more.

We first examine local context analysis. For queries improved by local context analysis, the average number of relevant passages in the top-ranked set is 29.7 per query. For queries degraded, the number is 8.6. For queries with 18 or more relevant passages, none of them is hurt by local context analysis. For queries with fewer than 18 relevant passages, some are hurt, and some are improved by local context analysis.

We now examine local feedback. The improved queries have 32 relevant passages in the top-ranked set per query on average, while the degraded queries have 20 relevant passages per query. This analysis shows that local feedback generally requires more relevant passages in the top-ranked set to improve a query. Unlike local context analysis, local feedback hurts a number of queries even if they have a substantial number of relevant passages in the top-ranked set. An examination of the top-ranked passages shows that such queries usually contain a large cluster of nonrelevant passages. This supports our hypothesis that local feedback fails if the relevant cluster is not the largest in the top-ranked set. One such query is "What research is ongoing to reduce the effects of osteoporosis in existing patients as well as prevent the disease occurring in those unafflicted at this time?" Many passages about treatment of heart disease are retrieved because they happen to use many of the query terms. Because local feedback selects expansion features based on frequency in the top-ranked set, terms such as "cholesterol" and "coronary" are chosen for query expansion. In comparison, local context analysis avoids this problem because the passages about heart disease do not use the query term "osteoporosis," and therefore concepts in them will not cooccur with that the term.

The fact that local context analysis is still dependent on the relevance of some top-ranked passages/documents means that if we can predict with

high accuracy whether the top-ranked set contains enough relevant passages/documents, we would be able to do a better job improving retrieval performance. Rather than expand all queries, we would only expand those which are likely to have enough top-ranked relevant passages. One method we have tried is based on the number of matched query terms in the top-ranked passages. The basic idea is that if the top-ranked passages contain few query terms, they are unlikely to be relevant, and we will not expand the query. Unfortunately, this simple method does not work. Data obtained on TREC4 show that the queries improved and show that the queries hurt by local context analysis have roughly the same average number of matched query terms (about 4.0) in a top-ranked passage when the top 10 passages are considered for each query. Whether more refined methods will work awaits further investigation.

8.3 Differences in Expansion Features

We found that local context analysis and local feedback select very different expansion features, even though both techniques operate on the top-ranked set. When the best runs of the two techniques on TREC4 are considered, the average number of unique words in the expansion features is 58 per query for local feedback and 78 for local context analysis (multiword features are broken into single words in computing the statistics). The number of overlapping words is only 17.6 per query. Some queries expanded quite differently are improved by both methods. The small overlap is understandable because the expansion features chosen by local feedback and local context analysis have different properties. In general, those selected by local feedback have a high frequency in the top-ranked set while those selected by local context analysis cooccur with all query terms in the top-ranked set.

9. COMPARING LOCAL CONTEXT ANALYSIS WITH PHRASEFINDER

We have discussed the impact of the number of passages used on the performance of local context analysis. We now revisit this issue. If all passages in a collection are used, local context analysis will be very similar to Phrasefinder in that both techniques prefer expansion concepts that cooccur with all query terms. In other words, Phrasefinder can be viewed as an extreme case of local context analysis. The performance curve of local context analysis in Figure 2 predicts that this extreme case of local context analysis will be worse than using the top-ranked passages. This is confirmed by the actual retrieval results of Phrasefinder.

The inferior performance of Phrasefinder is due to the low-ranked passages (documents). Since the overwhelming majority (approaching 100% on large collections) of them are not even remotely related to the topic of the query, using them is not only inefficient, but also hurts the chance of choosing the good concepts for query expansion. For example, let us consider one TREC4 query “Is there data available to suggest that capital punishment is a deterrent to crime?” Phrasefinder added some nonrelevant

Table XI. Effect of Passage Size and Number of Passages Used on Retrieval Performance of Local Context Analysis on WEST

Passage Size	Number of Passages Used							
	10	20	30	40	50	100	200	300
100	55.3 +2.8	55.9 +3.8	55.8 +3.7	56.3 +4.7	56.6 +5.2	55.4 +3.0	54.8 +1.8	54.7 +1.6
200	55.3 +2.8	56.7 +5.5	57.0 +5.9	57.3 +6.5	57.0 +6.0	56.1 +4.4	55.1 +2.5	54.6 +1.5
300	55.9 +3.8	56.5 +5.0	55.6 +3.4	55.7 +3.6	55.8 +3.7	55.6 +3.3	54.6 +1.6	54.4 +1.2
400	56.5 +4.9	56.7 +5.4	55.7 +3.6	55.5 +3.1	55.8 +3.8	55.9 +4.0	54.5 +1.3	53.7 −0.1
500	56.0 +4.1	56.4 +4.8	56.3 +4.7	56.7 +5.5	56.2 +4.5	55.9 +4.0	55.0 +2.2	54.2 +0.7

concepts such as “Iraqi leader,” “world order,” “Iraqi army,” and “shields” to the query because they cooccur with query terms “available,” “suggest,” “capital,” “crime,” and “deterrent” in the collection. It happens that many documents in the collection are about the military operation called Desert Storm, and many of the terms in the query occur in some of these documents (though not simultaneously). Since Phrasefinder uses global cooccurrences for query expansion, it chooses the common concepts from these documents. This is not a problem for local context analysis because these passages are not in the top-ranked set.

10. PARAMETER VARIATION

We now experiment with different parameter values and see how the performance of local context analysis is affected. The parameters we consider are the passage size, the δ-value (in the function $f(c, Q)$), and the number of concepts added to a query.

10.1 Passage Size

In Table XI, we list the retrieval performance of local context analysis on WEST using different passage sizes. We experimented with passage sizes 100, 200, 300, 400, and 500 words. For each passage size, we used the top 10, 20, 30, 40, 50, 100, 200, and 300 passages for query expansion. As we can see from the table, local context analysis produces improvements over the baseline for a wide range of passage sizes and numbers of passages used. In general, with a larger passage size, optimal performance occurs with a smaller number of passages. Performance seems to be independent of the passage size.

Though the experiments on WEST show that the performance is independent of the passage size, we believe that neither too large a passage size

Table XII. Effect of δ-Values on Performance of Local Context Analysis on TREC4

	Precision (% change)—49 Queries					
Recall	0.001	0.01	0.05	0.1	0.2	0.3
0	77.8	75.7 (−2.7)	73.4 (−5.6)	73.2 (−5.9)	72.9 (−6.3)	73.1 (−6.0)
10	55.1	54.9 (−0.2)	57.0 (+3.6)	57.1 (+3.7)	56.8 (+3.2)	56.6 (+2.7)
20	46.7	47.4 (+1.6)	47.7 (+2.2)	46.8 (+0.3)	46.6 (−0.1)	46.5 (−0.5)
30	40.0	40.4 (+1.0)	40.0 (+0.1)	39.9 (−0.2)	39.6 (−0.9)	39.4 (−1.5)
40	34.8	35.4 (+1.6)	35.4 (+1.7)	35.3 (+1.4)	34.8 (+0.1)	34.6 (−0.7)
50	29.4	30.1 (+2.4)	30.2 (+2.6)	29.9 (+1.7)	30.1 (+2.4)	30.1 (+2.2)
60	22.5	23.1 (+2.6)	23.7 (+5.3)	23.6 (+5.2)	24.1 (+7.3)	24.1 (+7.1)
70	15.9	17.0 (+6.6)	17.7 (+11.2)	17.9 (+12.3)	17.6 (+10.7)	17.6 (+10.3)
80	11.0	11.6 (+6.0)	11.7 (+7.2)	11.8 (+7.3)	11.9 (+8.3)	11.9 (+8.2)
90	5.1	5.3 (+4.6)	5.6 (+11.3)	5.7 (+11.9)	5.9 (+15.4)	5.9 (+16.8)
100	0.8	0.9 (+11.2)	0.8 (+7.5)	0.8 (+7.4)	0.9 (+11.9)	0.9 (+11.6)
Avg	30.8	31.1 (+0.8)	31.2 (+1.3)	31.1 (+0.9)	31.0 (+0.7)	30.9 (+0.4)

nor too small a passage size is desirable. If the passage size is too small, there will be fewer matched words between a query and the passages. A passage matching only the noncontent words in the query may be ranked high for the query. Consequently, the quality of the retrieved passages is affected. Since the WEST queries are short, this is not a serious problem. But this could be a problem for longer queries such as the TREC3 queries. If the passage size is too large, it will slow query expansion because the top-ranked passages may contain extraneous text. It seems that 300 words is a good choice for the passage size.

The best approach may be to segment long documents into passages so that each passage is about a topic. Techniques that automatically segment long documents or text streams by topics have been proposed in a number of studies [Hearst 1994; Ponte and Croft 1997]. We plan to investigate the application of these techniques for local context analysis in future work.

10.2 δ-Value

Table XII shows the effect of the δ-value on the performance of local context analysis on TREC4. We can see that the average precision is relatively insensitive to the δ-value. But precision at individual recall points is affected. In general, a small δ-value is good for precision, and a large δ-value is good for recall. We have discussed that a small δ-value favors concepts cooccurring with all query terms, while a large δ-value favors those cooccurring with individual query terms. The experimental results seem to imply that concepts cooccurring with all query terms are good for precision, and concepts cooccurring with individual query terms are good for recall.

10.3 Number of Concepts to Use

In the previous experiments, we added 70 concepts to each query. Adding so many concepts can significantly slow the retrieval process. We now

Table XIII. Using 70 Concepts versus Using 30 Concepts on TREC4

	Precision (% change)—49 Queries		
Recall	Baseline	lca-70-cpt	lca-30-cpt
0	71.0	73.2 (+3.2)	72.9 (+2.7)
10	49.3	57.1 (+15.7)	56.6 (+14.8)
20	40.4	46.8 (+16.0)	47.3 (+17.2)
30	33.3	39.9 (+19.8)	40.5 (+21.6)
40	27.3	35.3 (+29.1)	35.2 (+28.9)
50	21.6	29.9 (+38.4)	29.7 (+37.1)
60	14.8	23.6 (+59.8)	22.5 (+52.0)
70	9.5	17.9 (+89.1)	16.3 (+72.5)
80	6.2	11.8 (+91.0)	10.9 (+77.7)
90	3.1	5.7 (+80.2)	5.5 (+74.7)
100	0.4	0.8 (+88.2)	0.8 (+75.9)
Average	25.2	31.1 (+23.5)	30.7 (+22.1)

examine how the performance of local context analysis is affected if we use only 30 concepts for query expansion. A property of the INQUERY #WSUM operator is that the fewer operands that are inside the operator, the larger the belief value it returns. To offset the larger belief value produced by the auxiliary query because of the decrease in the number of concepts used, we set the weight of the auxiliary query to 1.0 rather than the default 2.0. The retrieval performance is shown in Table XIII. The performance using 30 concepts is only slightly worse than that of using 70 concepts. This means that the concepts ranked below 30 are only slightly useful for retrieval.

11. RELEVANCE FEEDBACK

For automatic query expansion where we do not know which retrieved documents are relevant, we have shown that the feature selection metric of local context analysis is better than that of local feedback. In a true relevance feedback environment where we know which retrieved documents are relevant, however, selecting the most frequent terms in the relevant documents is a better strategy than selecting terms based on cooccurrence, as shown by the experiments presented below.

We used 50 queries from the TREC5 and TREC6 topics to form a query set and used the Financial Times documents in TREC volume 4 as a training collection. For each query, 20 documents were retrieved from the training collection to form a training sample. Two versions of expanded queries were created. The first version was created by standard relevance feedback: the most frequent terms (excluding stopwords) from the relevant documents in the training sample were used for query expansion. The second version used the feature selection strategy of local context analysis. That is, the expansion terms in the second version were chosen based on their cooccurrences with the query terms in the relevant passages in the training sample (a relevant passage is a passage from a relevant document). For a fair comparison, both versions used the same number of

Table XIV. Comparing Term Selection by Frequency and Term Selection by Cooccurrence for Relevance Feedback

	Precision (% change)—50 Queries		
Recall	Base	Feedback by Frequency	Feedback by Cooccurrence
0	77.7	82.0 (+5.6)	82.9 (+6.8)
10	50.6	56.0 (+10.6)	56.9 (+12.2)
20	37.1	48.3 (+30.2)	48.4 (+30.4)
30	30.0	41.1 (+37.3)	37.9 (+26.7)
40	24.5	34.1 (+39.3)	31.0 (+26.9)
50	20.3	28.7 (+41.4)	24.8 (+22.2)
60	15.8	23.4 (+47.9)	19.8 (+25.6)
70	10.7	18.3 (+70.9)	15.4 (+44.0)
80	7.2	12.9 (+78.4)	10.6 (+47.1)
90	2.7	7.2 (+160.6)	6.4 (+133.4)
100	0.6	1.7 (+193.7)	1.9 (+225.7)
Average	25.2	32.1 (+27.6)	30.6 (+21.3)

expansion terms (30 per query) and the same weighting method (Rocchio $\alpha : \beta : \gamma = 2 : 8 : 1$). Two test collections were used: one for queries from TREC5 and consisting of AP and WSJ documents in TREC volume 2, and the other for queries from TREC6 and consisting of FBIS and the *Los Angeles Times* documents in TREC volume 5. Each query in the query set has more than 10 relevant documents in the training collection and in the test collection.

The results are shown in Table XIV. While both versions are significantly better than the unexpanded queries, standard relevance feedback is noticeably (about 5%) better than local context analysis. The results show that automatic query expansion without relevance judgments and true relevance feedback are quite different tasks and require different strategies for feature selection.

12. RESULTS ON CHINESE AND SPANISH COLLECTIONS

We have shown that local context analysis works on English collections. We now show that it also works on collections in other languages. The experiments in this section were carried out using the Chinese and Spanish collections of the TREC5 conference. They were mostly carried out by fellow IR researchers at UMass for the TREC5 conference. As a result, some parameter values are different from the ones used in the previous experiments.

The Chinese experiments were carried out on the TREC5-CHINESE collection. Query terms are Chinese words recognized by *Useg* [Ponte and Croft 1996], a Chinese segmenter based on the hidden Markov model. We define concepts as words in the documents recognized by the segmenter. Documents are broken into passages containing no more than 1500 Chinese characters. To expand a query, the top 30 concepts from the top 10 passages for the query are used. Concept i is assigned the weight

Table XV. Performance of Local Context Analysis on TREC5-CHINESE

	Precision (% change)—19 Queries	
Recall	Base	lca
0	69.1	74.9 (+8.4)
10	56.8	60.7 (+6.8)
20	49.2	56.2 (+14.1)
30	43.1	48.5 (+12.4)
40	37.2	44.2 (+18.9)
50	33.2	36.3 (+9.3)
60	26.9	31.2 (+16.1)
70	20.1	26.3 (+31.2)
80	16.8	21.9 (+30.1)
90	12.5	14.5 (+16.2)
100	3.7	5.7 (+53.4)
Average	33.5	38.2 (+14.0)

$$= 1.0 - (i - 1)/60.$$

The weight of the auxiliary query is set to 1.0. Table XV shows the retrieval performance of local context analysis on the Chinese collection. The improvement over the baseline is 14% and is statistically significant (p-value = 0.01).

The Spanish experiments were carried out on the TREC5-SPANISH collection. Concepts are noun phrases in the documents recognized by a part-of-speech tagger for Spanish. The passage size is 200 words. The top 31 concepts from the top 20 passages are added to each query. The top concept is given the weight 1.0 with all subsequent concepts downweighted by 1/100 for each position further down the rank. The weight of the auxiliary query is 1.0. Table XVI shows the retrieval performance of local context analysis on the Spanish collection. Local context analysis produces a 13% improvement in average precision over the unexpanded queries. The t-test indicates the improvement is statistically significant (p-value = 0.005). The precision at recall 0.0 is 7% worse, but the drop is not statistically significant. The drop appears to be caused by a few outliers in the query set. The small size of the query set (25 queries) makes its performance susceptible to outliers.

13. CROSS CORPORA EXPANSION

Local context analysis assumes that query words and their alternatives have some chance to cooccur in a collection. In other words, it assumes that vocabulary X and its alternative Y are used together in some documents D in a collection. When a user posts a query written in X, we search for it and find documents D. Then from documents D we find the alternative vocabulary Y and use it to expand the query. In fact, this is the assumption behind all query expansion techniques except for perhaps manual thesauri. But the assumption does not always hold. It sometimes happens that X and

Table XVI. Performance of Local Context Analysis on TREC5-SPANISH

	Precision (% change)—25 Queries	
Recall	Base	lca
0	85.5	79.6 (−7.0)
10	72.0	74.9 (+4.2)
20	55.2	66.8 (+21.1)
30	49.8	60.1 (+20.8)
40	45.4	51.2 (+12.9)
50	39.8	47.1 (+18.3)
60	33.5	40.5 (+20.8)
70	27.9	34.9 (+24.7)
80	22.0	28.2 (+28.1)
90	16.6	21.3 (+27.9)
100	1.8	3.5 (+86.9)
Average	40.9	46.2 (+13.0)

Y are not used together in any documents in a collection. For example, a reporter used the query "elderly black Americans" to search a collection of congressional bills and found no relevant documents for his query because politicians do not use "black Americans" to describe "African-Americans" [Croft et al. 1995].

One method to address the above problem is to expand a query on a different collection which indeed uses alternative vocabularies in its documents. A similar technique was used by several groups in TREC6 and TREC7: other sources of documents in addition to the original collection were used for query expansion [Allan et al. 1998; Kwok et al. 1998; Walker et al. 1998]. We have done an experiment to demonstrate this idea. The experiment was carried out on TREC5. But unlike previous experiments, the TREC5 queries were expanded on a different collection. The collection consists all the newspaper articles in TREC volumes 1–5, totaling 3GB, with sources Associated Press, *Wall Street Journal*, *San Jose Mercury*, *Financial Times*, Foreign Broadcast Information Service, and the *Los Angeles Times*. Since newspaper articles generally have a very wide audience, we conjecture that they would use different vocabularies and, therefore, be a good source of documents for query expansion. The conjecture is supported by the retrieval results in Table XVII. Using the newspaper collection significantly improves the retrieval performance (14.3% over the baseline and 11.8% over using the native TREC5 collection for query expansion).

14. VERY SHORT QUERIES

Although queries in some of the query sets used in previous sections are relatively short (see Table I), queries in some applications are even shorter. For example, applications that provide searching across the World Wide Web typically record average query lengths of two words [Croft et al. 1995]. In order to simulate retrieval in such applications, we did an experiment

Table XVII. Using a Newspaper Collection to Expand TREC5 Queries. One hundred passages are used.

	Precision (% change)—50 Queries		
Recall	Base	Use TREC5	Use Newspaper
0	64.1	53.7 (−16.2)	60.2 (−6.1)
10	37.5	34.4 (−8.4)	41.3 (+10.0)
20	29.1	30.9 (+6.3)	34.7 (+19.2)
30	24.1	26.4 (+9.3)	28.6 (+18.4)
40	21.3	23.5 (+10.5)	24.7 (+16.4)
50	17.9	20.7 (+15.8)	21.8 (+21.8)
60	12.6	17.1 (+36.2)	18.5 (+47.3)
70	10.1	12.7 (+25.2)	15.3 (+50.9)
80	7.2	8.7 (+21.6)	9.9 (+37.9)
90	4.8	6.2 (+28.2)	6.8 (+42.4)
100	2.7	2.4 (−13.5)	2.7 (+0.0)
Average	21.0	21.5 (+2.3)	24.1 (+14.3)

using a set of very short queries to search the TREC5 collections. The queries are the title fields of the TREC5 topics and mostly consist of a phrase, e.g., "gun control," computer security," and so forth. The average query length after removal of stopwords is only 3.2 words per query, which is close to the length of typical queries on the World Wide Web. In comparison, the average length of the TREC5 queries used in previous sections is 7.1 words per query. Since the queries are significantly shorter, word mismatch is a more serious problem. We conjecture that query expansion should produce more substantial improvement than using the longer queries.

The conjecture is supported by retrieval results in Table XVIII. The queries were expanded by local context analysis using the newspaper collection described in Section 13. Query expansion results in a substantial 24.9% improvement over the unexpanded queries. In comparison, query expansion results in smaller (14.3%) improvement for the long queries (Table XVII). Without query expansion, the short queries are 19.4% worse than the long queries. This suggests that for the same information need, word mismatch is more serious and query expansion more helpful for a short query than for a long query. We should note that the same is not necessarily true for different information needs, because the effectiveness of query expansion also depends on other factors. For example, the WEST queries are shorter than the TREC3 queries; however, the WEST baseline is much better than the TREC3 baseline, and query expansion is less helpful for the WEST queries than for the TREC3 queries. Further work is needed to find other factors affecting the effectiveness of query expansion.

15. EFFICIENCY AND OPTIMIZATION

In this section we discuss the computational costs of local context analysis. We should note that the main purpose of this article is to demonstrate the usefulness of the technique, and therefore efficiency is a secondary consid-

Table XVIII. Using Local Context Analysis to Expand TREC5 Title Queries

	Precision (% change)—50 Queries	
Recall	Title	Title-lca
0	53.2	57.5 (+8.1)
10	29.7	35.1 (+18.3)
20	23.7	29.3 (+23.4)
30	20.1	24.9 (+23.7)
40	17.5	21.9 (+25.3)
50	15.2	19.6 (+29.1)
60	10.8	16.5 (+53.1)
70	6.9	12.3 (+77.9)
80	4.9	8.4 (+71.0)
90	2.8	5.1 (+83.0)
100	1.8	2.5 (+40.5)
Average	17.0	21.2 (+24.9)

eration in our implementation. Before we present the performance figures under the current implementation, it is important to know the issues and overheads when local context analysis is used in a production system.

In a production environment, local context analysis would be an integral part of the retrieval system rather than separate software. The only augmentation to the index structure is a dictionary which stores the frequencies of the concepts in a collection. At indexing time, we need to recognize the concepts as documents are parsed. Based on our current implementation, we estimate that concept recognition will increase the indexing time by at most 50%. At query time, the system retrieves the top-ranked passages, parses them, collects the cooccurrences between concepts and query terms, and then ranks the concepts. Finally, the system adds the best concepts to the query and performs a second retrieval. Most of the extra work at query time is likely to be the initial retrieval. If parsing the top-ranked passages turns out to be the bottleneck, an option is to store the concepts in the index structure. That will slightly increase the storage overhead.

We now report the overheads under our prototype implementation. Experiments were carried out on a DEC alpha workstation. The TREC4 collection (2GB of text) was used in the experiments. Passage size is 300 words. Our implementation requires a local context analysis database in order to carry out query expansion on a collection. The database for TREC4 takes about 0.67GB, which is broken down as following:

—The concept dictionary, which stores the frequency of the concepts, 167MB.

—The term dictionary, which stores the frequency of the terms, 43MB.

—The concept file, which sequentially stores for each passage the concepts that occur in the passage and the numbers of occurrences, 251MB.

Table XIX. Filtering Low-Frequency (less than five times) Concepts on the Performance of Local Context Analysis (TREC4)

	Precision (% change)—49 Queries	
Recall	No Filter	Filter
0	73.2	71.4 (−2.4)
10	57.1	56.9 (−0.3)
20	46.8	46.9 (+0.0)
30	39.9	40.3 (+1.1)
40	35.3	35.1 (−0.4)
50	29.9	29.9 (−0.2)
60	23.6	23.2 (−1.9)
70	17.9	17.4 (−2.7)
80	11.8	12.0 (+1.7)
90	5.7	5.7 (−0.4)
100	0.8	0.9 (+1.1)
Average	31.1	30.9 (−0.7)

—The term file, which is analogous to the concept file except it is for terms, 213MB.

The time to build the local context analysis database for TREC4 is about four hours of wall clock time, most of which is spent on parsing and part-of-speech tagging. Currently, we also need to index the passages in order for INQUERY to retrieve the top-ranked passages for a query. The is simply a software artifact. With minor modification to the retrieval system, INQUERY can retrieve the top-ranked passages without creating a separate index for passages.

Query expansion consists of two steps. In the first step, INQUERY retrieves the passage identifiers of the top-ranked passages for a query. This step takes about 10 seconds per query. In the second step, concepts in the top-ranked passages are ranked, and the top-ranked concepts are output for query expansion. This steps takes about two seconds per query when 100 passages are used. The total time to expand a query is about 12 seconds.

The memory usage under the current implementation is very high (200MB), but can be easily cut to an acceptable level. The high memory usage is due to a large number of infrequent concepts in the concept dictionary. The TREC4 concept dictionary contains 4.9 million concepts, but most of them occur no more than a couple of times in the whole collection. Experimental results in Table XIX show that such concepts have limited, if any, impact on retrieval effectiveness. Filtering out those concepts occurring less than five times only affected retrieval performance by 0.7%. However, the size of the concept dictionary is reduced from 167MB to 17MB.

16. OTHER APPLICATIONS

Recently, we and fellow IR researchers at UMass have applied local context analysis in other IR-related problems and demonstrated promising results.

One application is distributed IR [Xu and Callan 1998]. A critical problem for distributed IR is choosing the right collections to search for a query. This is usually done by comparing a query with the dictionary information of each collection (e.g., terms and their document frequency in the collection). However, this method does not work well for typical ad hoc queries because most of the query terms are not discriminatory enough for the purpose of separating good collections from bad ones. Local context analysis can find highly specific terms for a query and enhance the discriminatory power of the query.

The second application is cross-language retrieval [Ballesteros 1997], where a query in one language must be translated in order to be used for retrieval in another language (e.g., English to Spanish). A major hindrance for effective cross-language retrieval is the poor translation caused by the ambiguity of the query terms. Local context analysis can provide very specific expansion concepts and as a result improve the quality of query translation and retrieval effectiveness.

The third application is topic segmentation [Ponte and Croft 1997]. The task of topic segmentation is to detect topic transitions in a text stream (e.g., a news feed) and break it into coherent documents. The commonly used technique is to compare two adjacent pieces of text (e.g., sentences) to see whether they share any words. The assumption is that within-topic sentences significantly share words, and cross-topic sentences do not. However, the assumption is often violated because of synonyms and word ambiguity. The solution is to treat the two pieces of text as two queries and expand them using local context analysis. Because the expansion concepts are related to the original texts, comparing the expanded texts results in more accurate detection of topic changes.

17. CONCLUSIONS AND FUTURE WORK

In this article we have proposed a new technique for automatic query expansion, called local context analysis. Experimental results on a number of collections show that it is more effective than existing techniques.

We will pursue the work in several directions. Firstly, the current function for concept selection, though it works well, is mostly heuristically driven. We hope that a function based on more formal considerations will further improve the retrieval performance. We are currently investigating several approaches including using language models for concept selection (J. Ponte, 1998, personal communications). Secondly, we need to be more flexible in query expansion. Currently we expand every query, even though some queries are inherently ambiguous and the best strategy is no expansion at all. An ambiguous query typically retrieves several clusters of documents which match the query equally well. We hope to utilize this property to determine whether a query is ambiguous. For an ambiguous query, we can choose to not expand it or ask the user to refine it. Lastly, our method to assign weights to the expansion concepts is ad hoc and needs to be improved.

ACKNOWLEDGMENTS

Thanks to Lisa Ballesteros, who did all the Spanish experiments, and John Broglio and Hongmin Shu, who did the major part of the Chinese experiments. We would like to thank the three reviewers. Their comments and suggestions greatly increase the quality of this article.

REFERENCES

ALLAN, J., CALLAN, J., CROFT, W., BALLESTEROS, L., BYRD, D., SWAN, R., AND XU, J. 1998. INQUERY does battle with TREC-6. In *Proceedings of the 6th Text Retrieval Conference* (TREC-6), E. Voorhees, Ed. 169–206. NIST Special Publication 500-240.

ATTAR, R. AND FRAENKEL, A. S. 1977. Local feedback in full-text retrieval systems. *J. ACM 24*, 3 (July), 397–417.

BALLESTEROS, L. AND CROFT, W. B. 1997. Phrasal translation and query expansion techniques for cross-langauge information retrieval. In *Proceedings of the 20th Annual International ACM Conference on Research and Development in Information Retrieval* (SIGIR '97, Philadelphia, PA, July 27–31), N. J. Belkin, A. D. Narasimhalu, P. Willett, W. Hersh, F. Can, and E. Voorhees, Eds, ACM Press, New York, NY, 84–91.

BROGLIO, J., CALLAN, J. P., AND CROFT, W. 1994. An overview of the INQUERY system as used for the TIPSTER project. In *Proceedings of the TIPSTER Workshop*, Morgan Kaufmann, San Mateo, CA, 47–67.

BROGLIO, J., CALLAN, J. P., CROFT, W. B., AND NACHBAR, D. W. 1995. Document retrieval and routing using the INQUERY system. In *Proceedings of the 3rd Text Retrieval Conference* (TREC-3), D. Harman, Ed. National Institute of Standards and Technology, Gaithersburg, MD, 22–29.

BUCKLEY, C., MITRA, M., WALZ, J., AND CARDIE, C. 1998. Using clustering and superconcepts within SMART. In *Proceedings of the 6th Text Retrieval Conference* (TREC-6), E. Voorhees, Ed. 107–124. NIST Special Publication 500-240.

BUCKLEY, C., SALTON, G., ALAN, J., AND SINGHAL, A. 1995a. Automatic query expansion using SMART. In *Proceedings of the 3rd Text Retrieval Conference* (TREC-3), D. Harman, Ed. National Institute of Standards and Technology, Gaithersburg, MD, 69–80.

BUCKLEY, C., SINGHAL, A., MITRA, M., AND SALTON, G. 1995b. New retrieval approaches using SMART. In *Proceedings of the 4th Text Retrieval Conference* (TREC-4, Washington, D.C., Nov.), D. K. Harman, Ed. National Institute of Standards and Technology, Gaithersburg, MD, 25–48.

CAID, W. R., DUMAIS, S. T., AND GALLANT, S. I. 1995. Learned vector-space models for document retrieval. *Inf. Process. Manage. 31*, 3 (May–June), 419–429.

CHURCH, K. W. AND HANKS, P. 1989. Word association norms, mutual information and lexicography. In *Proceedings of ACL 27* (Vancouver, Canada), 76–83.

CROFT, W. AND HARPER, D. J. 1979. Using probabilistic models of document retrieval without relevance information. *J. Doc. 35*, 285–295.

CROFT, W. B., COOK, R., AND WILDER, D. 1995. Providing government information on the Internet: Experiences with THOMAS. In *Proceedings of the 2nd International Conference on Theory and Practice of Digital Libraries* (DL '95, Austin, TX, June), 19–24.

DEERWESTER, S., DUMAI, S. T., FURNAS, G. W., LANDAUER, T. K., AND HARSHMAN, R. 1990. Indexing by latent semantic analysis. *J. Am. Soc. Inf. Sci. 41*, 6, 391–407.

FOX, E. A. 1983. Extending the Boolean and vector space models of information retrieval with P-norm queries and multiple concept types. Ph.D. Dissertation. Cornell University, Ithaca, NY.

FURNAS, G. W., DEERWESTER, S., DUMAIS, S. T., LANDAUER, T. K., HARSHMAN, R. A., STREETER, L. A., AND LOCHBAUM, K. E. 1988. Information retrieval using a singular value decomposition model of latent semantic structure. In *Proceedings of the 11th International Conference on Research and Development in Information Retrieval* (SIGIR '88, Grenoble, France, June 13–15), Y. Chiaramella, Ed. ACM Press, New York, NY, 465–480.

FURNAS, G. W., LANDAUER, T. K., GOMEZ, L. M., AND DUMAIS, S. T. 1987. The vocabulary problem in human-system communication. *Commun. ACM 30*, 11 (Nov. 1987), 964–971.

GREIFF, W. R. 1998. A theory of term weighting based on exploratory data analysis. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (SIGIR '98, Melbourne, Australia, Aug. 24–28), W. B. Croft, A. Moffat, C. J. van Rijsbergen, R. Wilkinson, and J. Zobel, Eds. ACM Press, New York, NY, 11–19.

HAWKING, D., THISTLEWAITE, P., AND CRASWELL, N. 1998. ANU/ACSys TREC-6 experiments. In *Proceedings of the 6th Text Retrieval Conference* (TREC-6), E. Voorhees, Ed. 275–290. NIST Special Publication 500-240.

HEARST, M. 1994. Mini-paragraph segmentation of expository discourse. In *Proceedings of the 32nd Meeting of the ACL*,

HEARST, M. A. AND PEDERSEN, J. O. 1996. Reexamining the cluster hypothesis: scatter/gather on retrieval results. In *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (SIGIR '96, Zurich, Switzerland, Aug. 18–22), H.-P. Frei, D. Harman, P. Schaüble, and R. Wilkinson, Eds. ACM Press, New York, NY, 76–84.

HULL, D. 1993. Using statistical testing in the evaluation of retrieval experiments. In *Proceedings of the 16th Annual International ACM Conference on Research and Development in Information Retrieval* (SIGIR '93, Pittsburgh, PA, June 27–July), R. Korfhage, E. Rasmussen, and P. Willett, Eds. ACM Press, New York, NY, 329–338.

JING, Y. AND CROFT, W. B. 1994. An association thesaurus for information retrieval. In *Proceedings of the Intelligent Multimedia Information Retrieval Systems* (RIAO '94, New York, NY), 146–160.

KWOK, K. L. 1996. A new method of weighting query terms for ad-hoc retrieval. In *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (SIGIR '96, Zurich, Switzerland, Aug. 18–22), H.-P. Frei, D. Harman, P. Schaüble, and R. Wilkinson, Eds. ACM Press, New York, NY, 187–195.

KWOK, K. L., GRUNFELD, L., AND XU, J. 1998. TREC-6 English and Chinese experiments using PIRCS. In *Proceedings of the 6th Text Retrieval Conference* (TREC-6), E. Voorhees, Ed. 207–214. NIST Special Publication 500-240.

LU, A., AYOUB, M., AND DONG, J. 1997. Ad hoc experiments using EUREKA. In *Proceedings of the 5th Text Retrieval Conference*, 229–240. NIST Special Pub 500-238.

MINKER, J., WILSON, G., AND ZIMMERMAN, B. 1972. An evaluation of query expansion by the addition of clustered terms for a document retrieval system. *Inf. Storage Retrieval 8*, 329–348.

MITRA, M., SINGHAL, A., AND BUCKLEY, C. 1998. Improving automatic query expansion. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (SIGIR '98, Melbourne, Australia, Aug. 24–28), W. B. Croft, A. Moffat, C. J. van Rijsbergen, R. Wilkinson, and J. Zobel, Eds. ACM Press, New York, NY, 206–214.

PONTE, J. M. AND CROFT, W. B. 1998. A language modeling approach to information retrieval. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (SIGIR '98, Melbourne, Australia, Aug. 24–28), W. B. Croft, A. Moffat, C. J. van Rijsbergen, R. Wilkinson, and J. Zobel, Eds. ACM Press, New York, NY, 275–281.

PONTE, J. AND CROFT, B. 1996. USeg: A retargetable word segmentation procedure for information retrieval. In *Proceedings of the Symposium on Document Analysis and Information Retrieval*,

PONTE, J. AND CROFT, B. 1997. Text segmentation by topic. In *Proceedings of the 1st European Conference on Research and Advanced Technology for Digital Libraries*, 113–125.

QIU, Y. AND FREI, H.-P. 1993. Concept based query expansion. In *Proceedings of the 16th Annual International ACM Conference on Research and Development in Information Retrieval* (SIGIR '93, Pittsburgh, PA, June 27–July), R. Korfhage, E. Rasmussen, and P. Willett, Eds. ACM Press, New York, NY, 160–169.

ROCCHIO, J. 1971. Relevance feedback in information retrieval. In *The Smart Retrieval System—Experiments in Automatic Document Processing*, G. Salton, Ed. Prentice-Hall, Englewood Cliffs, NJ, 313–323.

SALTON, G. 1989. *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer*. Addison-Wesley Series in Computer Science. Addison-Wesley Longman Publ. Co., Inc., Reading, MA.

SALTON, G. AND BUCKLEY, C. 1990. Improving retrieval performance by relevance feedback. *J. Am. Soc. Inf. Sci. 41*, 4, 288–297.

SCHÜTZE, H. AND PEDERSEN, J. 1994. A cooccurrence-based thesaurus and two applications to information retrieval. In *Proceedings of the Intelligent Multimedia Information Retrieval Systems* (RIAO '94, New York, NY), 266–274.

SINGHAL, A., BUCKLEY, C., AND MITRA, M. 1996. Pivoted document length normalization. In *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (SIGIR '96, Zurich, Switzerland, Aug. 18–22), H.-P. Frei, D. Harman, P. Schaüble, and R. Wilkinson, Eds. ACM Press, New York, NY, 21–29.

SPARCK JONES, K. 1971. *Automatic Keyword Classification for Information Retrieval*. Butterworths, London, UK.

VAN RIJSBERGEN, C. J. 1979. *Information Retrieval*. 2nd ed. Butterworths, London, UK.

VOORHEES, E. AND HARMAN, D. 1998. Overview of the Sixth Text Retrieval Conference (TREC-6). In *Proceedings of the 6th Text Retrieval Conference* (TREC-6), E. Voorhees, Ed. 1–24. NIST Special Publication 500-240.

WALKER, S., ROBERTSON, S., BOUGHANEM, M., JONES, G., AND JONES, K. S. 1997. Okapi at TREC-6 automatic ad hoc, VLC, routing, filtering and QSDR. In *Proceedings of the 6th Text Retreival Conference* (TREC-6, Nov.), E. Voorhees and D. Harman, Eds. 125–136.

WILKINSON, R., ZOBEL, J., AND SACKS-DAVIS, R. 1996. Similarity measures for short queries. In *Proceedings of the 4th Text Retrieval Conference*, D. Harman, Ed. 277–286. NIST Special Publication 500-236.

XU, J. 1997. Solving the word mismatch problem through automatic text analysis. Ph.D. Dissertation. Computer and Information Science Department, University of Massachusetts, Amherst, MA.

XU, J. AND CALLAN, J. 1998. Effective retrieval with distributed collections. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (SIGIR '98, Melbourne, Australia, Aug. 24–28), W. B. Croft, A. Moffat, C. J. van Rijsbergen, R. Wilkinson, and J. Zobel, Eds. ACM Press, New York, NY, 112–120.

XU, J. AND CROFT, W. B. 1996. Query expansion using local and global document analysis. In *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (SIGIR '96, Zurich, Switzerland, Aug. 18–22), H.-P. Frei, D. Harman, P. Schaüble, and R. Wilkinson, Eds. ACM Press, New York, NY, 4–11.

XU, J., BROGLIO, J., AND CROFT, B. 1994. The design and implementation of a part of speech tagger for English. Tech. Rep. IR52. Computer and Information Science Department, University of Massachusetts, Amherst, MA.

Received: June 1998; revised: February 1999 and December 1999; accepted: December 1999