Query Expansion Using Local and Global Document Analysis

Jinxi Xu and W. Bruce Croft
Center for Intelligent Information Retrieval
Computer Science Department
University of Massachusetts, Amherst
Amherst, MA 01003-4610, USA
xu@cs.umass.edu croft@cs.umass.edu

Abstract

Automatic query expansion has long been suggested as a technique for dealing with the fundamental issue of word mismatch in information retrieval. A number of approaches to expansion have been studied and, more recently, attention has focused on techniques that analyze the corpus to discover word relationships (global techniques) and those that analyze documents retrieved by the initial query (local feedback). In this paper, we compare the effectiveness of these approaches and show that, although global analysis has some advantages, local analysis is generally more effective. We also show that using global analysis techniques, such as word context and phrase structure, on the local set of documents produces results that are both more effective and more predictable than simple local feedback.

1 Introduction

The problem of word mismatch is fundamental to information retrieval. Simply stated, it means that people often use different words to describe concepts in their queries than authors use to describe the same concepts in their documents. The severity of the problem tends to decrease as queries get longer, since there is more chance of some important words co-occurring in the query and relevant documents. In many applications, however, the queries are very short. For example, applications that provide searching across the World-Wide Web typically record average query lengths of two words [Croft et al., 1995]. Although this may be one extreme in terms of IR applications, it does indicate that most IR queries are not long and that techniques for dealing with word mismatch are needed.

An obvious approach to solving this problem is query expansion. The query is expanded using words or phrases with similar meaning to those in the query and the chances of matching words in relevant documents are therefore increased. This is the basic idea behind the use of a thesaurus in query formulation. There is, however, little evidence that a general thesaurus is of any use in improving the effectiveness of the search, even if words are selected by the searchers [Voorhees, 1994]. Instead, it has been proposed that by automatically analyzing the text of the corpus being searched, a more effective thesaurus or query expansion technique could be produced.

One of the earliest studies of this type was carried out by Sparck Jones [Sparck Jones, 1971] who clustered words based on co-occurrence in documents and used those clusters to expand the queries. A number of similar studies followed but it was not until recently that consistently positive results have been obtained. The techniques that have been used recently can be described as being based on either global or local analysis of the documents in the corpus being searched. The global techniques examine word occurrences and relationships in the corpus as a whole, and use this information to expand any particular query. Given their focus on analyzing the corpus, these techniques are extensions of Sparck Jones' original approach.

Local analysis, on the other hand, involves only the top ranked documents retrieved by the original query. We have called it *local* because the techniques are variations of the original work on local feedback [Attar & Fraenkel, 1977, Croft & Harper, 1979]. This work treated local feedback as a special case of relevance feedback where the top ranked documents were assumed to be relevant. Queries were both reweighted and expanded based on this information.

Both global and local analysis have the advantage of expanding the query based on all the words in the query. This is in contrast to a thesaurus-based approach where individual words and phrases in the query are expanded and word ambiguity is a problem. Global analysis is inherently more expensive than local analysis. On the other hand, global analysis provides a thesaurus-like resource that can be used for browsing without searching, and retrieval results with local feedback on small test collections were not promising.

More recent results with the TREC collection, however, indicate that local feedback approaches can be effective and, in some cases, outperform global analysis techniques. In this paper, we compare these approaches using different query sets and corpora. In addition, we propose and evaluate a new technique which borrows ideas from global analysis, such as the use of context and phrase structure, but applies them to the *local* document set. We call the new technique local context analysis to distinguish it from local feedback.

In the next section, we describe the global analysis procedure used in these experiments, which is the *Phrasefinder* component of the INQUERY retrieval system [Jing & Croft,

SIGIR'96, Zurich, Switzerland©1996 ACM 0-89791-792-8/96/08 $3.50

1994]. Section 3 covers the local analysis procedures. The local feedback technique is based on the most successful approaches from the recent TREC conference [Harman, 1996]. Local context analysis is described in detail.

The experiments and results are presented in section 4. Both the TREC [Harman, 1995] and WEST [Turtle, 1994] collections are used in order to compare results in different domains. A number of experiments with local context analysis are reported to show the effect of parameter variations on this new technique. The other techniques are run using established parameter settings. In the comparison of global and local techniques, both recall/precision averages and query-by-query results are used. The latter evaluation is particularly useful to determine the robustness of the techniques, in terms of how many queries perform substantially worse after expansion. In the final section, we summarize the results and suggest future work.

2 Global Analysis

The global analysis technique we describe here has been used in the INQUERY system in TREC evaluations and other applications [Jing & Croft, 1994, Callan et al., 1995], and was one of the first techniques to produce consistent effectiveness improvements through automatic expansion. Other researchers have developed similar approaches [Qiu & Frei, 1993, Schütze & Pedersen, 1994] and have also reported good results.

The basic idea in global analysis is that the global context of a *concept* can be used to determine similarities between *concepts*. Context can be defined in a number of ways, as can concepts. The simplest definitions are that all words are concepts (except perhaps stop words) and that the context for a word is all the words that co-occur in documents with that word. This is the approach used by [Qiu & Frei, 1993], and the analysis produced is related to the representations generated by other dimensionality-reduction techniques [Deerwester et al., 1990, Caid et al., 1993]. The essential difference is that global analysis is only used for query expansion and does not replace the original word-based document representations. Reducing dimensions in the document representation leads to problems with precision. Another related approach uses clustering to determine the context for document analysis [Crouch & Yang, 1992].

In the Phrasefinder technique used with INQUERY, the basic definition for a concept is a noun group, and the context is defined as the collection of fixed length *windows* surrounding the concepts. A noun group (phrase) is either a single noun, two adjacent nouns or three adjacent nouns. Typical effective window sizes are from 1 to 3 sentences. One way of visualizing the technique, although not the most efficient way of implementing it, is to consider every concept (noun group) to be associated with a *pseudo-document*. The contents of the pseudo-document for a concept are the words that occur in every window for that concept in the corpus. For example, the concept *airline pilot* might have the words *pay*, *strike*, *safety*, *air*, *traffic* and *FAA* occurring frequently in the corresponding pseudo-document, depending on the corpus being analyzed. An INQUERY database is built from these pseudo-documents, creating a concept database. A filtering step is used to remove words that are too frequent or too rare, in order to control the size of the database.

To expand a query, it is run against the concept database using INQUERY, which will generate a ranked list of phrasal concepts as output, instead of the usual list of document names. Document and collection-based weighting of matching words are used to determine the concept ranking, in a similar way to document ranking. Some of the top-ranking phrases from the list are then added to the query and weighted appropriately. In the Phrasefinder queries used in this paper, 30 phrases are added into each query and are downweighted in proportion to their rank position. Phrases containing only terms in the original query are weighted more heavily than those containing terms not in the original query.

Figure 1 shows the top 30 concepts retrieved by Phrasefinder for the TREC4 query 214 "What are the different techniques used to create self induced hypnosis". While some of the concepts are reasonable, others are difficult to understand. This is due to a number of spurious matches with noncontent words in the query.

The main advantages of a global analysis approach like the one used in INQUERY is that it is relatively robust in that average performance of queries tends to improve using this type of expansion, and it provides a thesaurus-like resource that can be used for browsing or other types of concept search. The disadvantages of this approach is that it can be expensive in terms of disk space and computer time to do the global context analysis and build the searchable database, and individual queries can be significantly degraded by expansion.

3 Local Analysis

3.1 Local Feedback

The general concept of local feedback dates back at least to a 1977 paper by Attar and Fraenkel [Attar & Fraenkel, 1977]. In this paper, the top ranked documents for a query were proposed as a source of information for building an automatic thesaurus. Terms in these documents were clustered and treated as quasi-synonyms. In [Croft & Harper, 1979], information from the top ranked documents is used to re-estimate the probabilities of term occurrence in the relevant set for a query. In other words, the weights of query terms would be modified but new terms were not added. This experiment produced effectiveness improvements, but was only carried out on a small test collection.

Experiments carried out with other standard small collections did not give promising results. Since the simple version of this technique consists of adding common words from the top-ranked documents to the original query, the effectiveness of the technique is obviously highly influenced by the proportion of relevant documents in the high ranks. Queries that perform poorly and retrieve few relevant documents would seem likely to perform even worse after local feedback, since most words added to the query would come from non-relevant documents.

In recent TREC conferences, however, simple local feedback techniques appear to have performed quite well. In this paper, we expand using a procedure similar to that used by the Cornell group in TREC 4 & 3 [Buckley et al., 1996]. The most frequent 50 terms and 10 phrases (pairs of adjacent non stop words) from the top ranked documents are added to the query. The terms in the query are reweighted using the Rocchio formula with $\alpha : \beta : \gamma = 1 : 1 : 0$.

Figure 2 shows terms and phrases added by local feedback to the same query used in the previous section. In this case, the terms in the query are stemmed.

One advantage of local feedback is that it can be relatively efficient to do expansion based on high ranking documents. It may be slightly slower at run-time than, for

hypnosis	meditation	practitioners
dentists	antibodies	disorders
psychiatry	immunodeficiency-virus	anesthesia
susceptibility	therapists	dearth
atoms	van-dyke	self
confession	stare	proteins
katie	johns-hopkins-university	growing-acceptance
reflexes	voltage	ad-hoc
correlation	conde-nast	dynamics
ike	illnesses	hoffman

Figure 1: Phrasefinder concepts for TREC4 query 214

hypnot	hypnotiz	19960500
psychosomat	psychiatr	immun
mesmer	franz	suscept
austrian	dyck	psychiatrist
shesaid	tranc	professor
hallucin	18th	centur
hilgard	11th	unaccept
19820902	syndrom	exper
physician	told	patient
hemophiliac	strang	cortic
ol	defic	muncie
spiegel	diseas	imagin
suggest	dyke	feburar
immunoglobulin	reseach	fresco
person	numb	katie
psorias	treatment	medicin
17150000	ms	franz-mesmer
austrian-physician	psychosomat-medicin	
hypnot-state	fight-immun	intern-congress
late-18th	diseas-fight	hypnotiz-peopl
ms-ol		

Figure 2: Local feedback terms and phrases for TREC4 query 214

example, Phrasefinder, but needs no thesaurus construction phase. Local feedback requires an extra search and access to document information. If document information is stored only for this purpose, then this should be counted as a space overhead for the technique, but it likely to be significantly less than a concept database. A disadvantage currently is that it is not clear how well this technique will work with queries that retrieve few relevant documents.

3.2 Local Context Analysis

Local context analysis is a new technique which combines global analysis and local feedback. Like Phrasefinder, noun groups are used as concepts and concepts are selected based on co-occurrence with query terms. Concepts are chosen from the top ranked documents, similar to local feedback, but the best passages are used instead of whole documents. The standard INQUERY ranking is not used in this technique.

Below are the steps to use local context analysis to expand a query Q on a collection.

1. Use a standard IR system (INQUERY) to retrieve the top n ranked passages. A passage is a text window of fixed size (300 words in these experiments [Callan, 1994]).

   There are two reasons that we use passages rather than documents. Since documents can be very long and about multiple topics, a co-occurrence of a concept at the beginning and a term at the end of a long document may mean nothing. It is also more efficient to use passages because we can eliminate the cost of processing the unnecessary parts of the documents.

2. Concepts (noun phrases) in the top n passages are ranked according to the formula

$$bel(Q,c) = \prod_{t_i \in Q} (\delta + \log(af(c,t_i))\ idf_c / \log(n))^{idf_i}$$

Where

$$\begin{aligned} af(c,t_i) &= \sum_{j=1}^{j=n} ft_{ij}\ fc_j \\ idf_i &= max(1.0, log10(N/N_i)/5.0) \\ idf_c &= max(1.0, log10(N/N_c)/5.0) \end{aligned}$$

c	is a concept
ft_{ij}	is the number of occurrences of t_i in p_j
fc_j	is the number of occurrences of c in p_j
N	is the number of passages in the collection
N_i	is the number of passages containing t_i
N_c	is the number of passages containing c
δ	is 0.1 in this paper to avoid zero bel value

The above formula is a variant of the $tf\ idf$ measure used by most IR systems. In the formula, the af part

rewards concepts co-occurring frequently with query terms, the idf_c part penalizes concepts occurring frequently in the collection, the idf_i part emphasizes infrequent query terms. Multiplication is used to emphasize co-occurrence with all query terms.

3. Add m top ranked concepts to Q using the following formula:

$$Q_{new} = \#WSUM(1.0\ 1.0\ Q\ w\ Q\prime)$$
$$Q\prime = \#WSUM(1.0\ w_1\ c_1\ w_2\ c_2\ \ldots\ w_m\ c_m)$$

In our experiments, m is set to 70 and w_i is set to $1.0 - 0.9 * i/70$. Unless specified otherwise, w is set to 2.0. We call $Q\prime$ the auxiliary query. $\#WSUM$ is an INQUERY query operator which computes a weighted average of its components.

Figure 3 shows the top 30 concepts added by local context analysis to TREC4 query 214.

Local context analysis has several advantages. It is computationally practical. For each collection, we only need a single pass to collect the collection frequencies for the terms and noun phrases. This pass takes about 3 hours on an Alpha workstation for the TREC4 collection. The major overhead to expand a query is an extra search to retrieve the top ranked passages. On a modern computer system, this overhead is reasonably small. Once the top ranked passages are available, query expansion is fast: when 100 passages are used, our current implementation requires only several seconds of CPU time to expand a TREC4 query. So local context analysis is practical even for interactive applications. For queries containing proximity constraints (e.g. phrases), Phrasefinder may add concepts which co-occur with all query terms but do not satisfy proximity constraints. Local context analysis does not have such a problem because the top ranked passages are retrieved using the original query. Because it does not filter out frequent concepts, local context analysis also has the advantage of using frequent but potentially good expansion concepts. A disadvantage of local context analysis is that it may require more time to expand a query than Phrasefinder.

4 Experiments

4.1 Collections and Query Sets

Experiments are carried out on 3 collections: TREC3 that comprises Tipster 1 and 2 datasets with 50 queries (topics 151-200), TREC4 that comprises Tipster 2 and 3 datasets with 49 queries (topics 202-250) and WEST with 34 queries. TREC3 and TREC4 (about 2 GBs each) are much larger and more heterogeneous than WEST. The average document length of the TREC documents is only 1/7 of that of the WEST documents. The average number of relevant documents per query with the TREC collections is much larger than that of WEST. Table 1 lists some statistics about the collections and the query sets. Stop words are not included.

4.2 Local Context Analysis

Table 2 shows the performance of local context analysis on the three collections. 70 concepts are added into each query using the expansion formula in section 3.2.

Local text analysis performs very well on TREC3 and TREC4. All runs produce significant improvements over the baseline on the TREC collections. The best run on TREC4 (100 passages) is 23.5% better than the baseline. The best run on TREC3 (200 passages) is 24.4% better than the baseline. On WEST, the improvements over the baseline are not as good as on TREC3 and TREC4. With too many passages, the performance is even worse than the baseline.

The high baseline of the WEST collection (53.8% average precision) suggests that the original queries are of very good quality and we should give them more emphasis. So we downweight the expansion concepts by 50% by reducing the weight of auxiliary query $Q\prime$ from 2.0 to 1.0. Table 3 shows that downweighting the expansion concepts does improve performance.

It is interesting to see how the number of passages used affects retrieval performance. To see it more clearly, we plot the performance curve on TREC4 in figure 4. Initially, increasing the number of passages quickly improves performance. The performance peaks at a certain point. After staying relatively flat for a period, the performance curves drop slowly when more passages are used. For TREC3 and TREC4, the optimal number of passages is around 100, while on WEST, the optimal number of passages is around 20. This is not surprising because the first two collections are a order of magnitude larger than WEST. Currently we do not know how to automatically determine the optimal number of passages to use. Fortunately, local context analysis is relatively insensitive to the number of the passages used, especially for large collections like the TREC collections. On the TREC collections, between 30 and 300 passages produces very good retrieval performance.

5 Local Text Analysis vs Global Analysis

In this section we compare Phrasefinder and local context analysis in term of retrieval performance. Tables 4-5 compare the retrieval performance of the two techniques on the TREC collections. On both collections, local context analysis is much better than Phrasefinder. On TREC3, Phrasefinder is 7.8% better than the baseline while local context analysis using the top ranked 100 passages is 23.3% better than the baseline. On TREC4, Phrasefinder is only 3.4% better than the baseline while local context analysis using the top ranked 100 passages is 23.5% than the baseline. In fact, all local context analysis runs in table 2 are better than Phrasefinder on TREC3 and TREC4. On both collections, Phrasefinder hurts the high-precision end while local context analysis helps improve precision. The results show that local context analysis is a better query expansion technique than Phrasefinder.

We examine two TREC4 queries to show why Phrasefinder is not as good as local context analysis. For one example, "China" and "Iraq" are very good concepts for TREC4 query "Status of nuclear proliferation treaties – violations and monitoring". They are added into the query by local context analysis but not by Phrasefinder. It appears that they are filtered out by Phrasefinder because they are frequent concepts. For the other example, Phrasefinder added the concept "oil spill" to TREC4 query "As a result of DNA testing, are more defendants being absolved or convicted of crimes". This seems to be strange. It appears that Phrasefinder did this because "oil spill" co-occurs with many of the terms in the query, e.g., "result", "test", "defendant", "absolve" and "crime". But "oil spill" does not co-occur with "DNA", which is a key element of the query. While it is very hard to automatically determine which terms are key elements of a query, the product function used by local context analysis for selecting expansion concepts should be

hypnosis	brain-wave	ms.-burns
technique	pulse	reed
brain	ms.-olness	trance
hallucination	process	circuit
van-dyck	behavior	suggestion
case	spiegel	finding
hypnotizables	subject	van-dyke
patient	memory	application
katie	muncie	approach
study	point	contrast

Figure 3: Local Context Analysis concepts for query 214

collection	WEST	TREC3	TREC4
Number of queries	34	50	49
Raw text size in gigabytes	0.26	2.2	2.07
Number of documents	11,953	741,856	567,529
Mean words per document	1,970	260	299
Mean relevant documents per query	29	196	133
Number of words in a collection	23,516,042	192,684,738	169,682,351

Table 1: Statistics on text corpora

	Number of passages										
collection	10	20	30	40	50	100	200	300	500	1000	2000
TREC4	29.5	29.9	30.2	30.3	30.4	31.1	31.0	30.7	29.9	29.0	27.9
	+17	+18.6	+19.8	+20.3	+20.6	+23.5	+23.0	+21.8	+18.6	+15	+10.7
TREC3	36.6	37.5	38.7	39.0	38.9	38.9	39.3	39.1	38.3	37.6	36.6
	+16	+18.9	+22.6	+23.6	+23.2	23.3	+24.4	+23.7	+21.3	+19	+16.0
WEST	54.8	55.4	54.5	54.6	54.2	54.2	53.1	52.7	52.1	51.7	51.7
	+1.9	+3.0	+1.3	+1.6	+0.7	+0.8	-1.3	-2.0	-3.2	-3.9	-3.9

Table 2: Performance of local context analysis using 11 point average precision

	Number of passages										
collection	10	20	30	40	50	100	200	300	500	1000	2000
WEST	55.9	56.5	55.6	55.7	55.8	55.6	54.6	54.4	53.6	53.7	53.7
	+3.8	+5.0	+3.4	+3.6	+3.7	+3.3	+1.6	+1.2	-0.4	-0.1	-0.1

Table 3: Downweight expansion concepts of local context analysis on WEST. The weight of the auxiliary query is reduced to 1.0

better than the sum function used by Phrasefinder because with the product function it is harder for some query terms to dominate other query terms.

6 Local Text Analysis vs Local Feedback

In this section we compare the retrieval performances of local feedback and local context analysis. Table 7 shows the retrieval performance of local feedback.

Table 8 shows the result of downweighting the expansion concepts by 50% on WEST. The reason for this is to make a fair comparison with local context analysis. Remember that we also downweighted the expansion concepts of local context analysis by 50% on WEST.

Local feedback does very well on TREC3. The best run produces a 20.5% improvement over the baseline, close to the 24.4% of the best run of local context analysis. It is also relatively insensitive to the number of documents used for feedback on TREC3. Increasing the number of documents from 10 to 50 does not affect performance much.

It also does well on TREC4. The best run produces a 14.0% improvement over the baseline, very significant, but lower than the 23.5% of the best run of local context analysis. It is very sensitive to the number of documents used for feedback on TREC4. Increasing the number of documents from 5 to 20 results in a big performance loss. In contrast, local context analysis is relatively insensitive to the number of passages on all three collections.

On WEST, local feedback does not work at all. Without downweighting the expansion concepts, it results in a significant performance loss over all runs. Downweighting the expansion concepts only reduces the amount of loss. It is also sensitive to the number of documents used for feedback. Increasing the number of feedback documents results in significantly more performance loss.

It seems that the performance of local feedback and its sensitivity to the number of documents used for feedback depend on the number of relevant documents in the collection for the query. From table 1 we know that average number of relevant documents per query on TREC3 is 196, larger than 133 of TREC4, which is in turn larger than 29 of WEST. This corresponds to the relative performance of local feedback on the collections.

Tables 4-6 show a side by side comparison between local feedback and local context analysis at different recall levels on the three collections. Top 10 documents are used for local

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS
Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS
Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS
Sync your system to PACER to automate legal marketing.