Query Expansion on a Corporate Intranet: Using LSI to Increase Precision in Explorative Search

Dick Stenmark, PhD
Göteborg University
Department of informatics
Göteborg, Sweden
stenmark@informatik.gu.se

Abstract

Previous research has taught us that the typical non-professional information seeker on the World Wide Web submits very short queries resulting in low-precision results. We show that this behaviour is repeated also by intranet users and therefore apply query expansion (QE) techniques to improve their search results. Arguing that casual searchers are likely to be unwilling to engage in the dialogue required for interactive QE, we provide an automatic QE system based on Latent Semantic Indexing (LSI). Having received mixed results, our analysis suggests that automatic QE based on a collection dependent knowledge structure may work for explorative, i.e. broader, queries whilst targeted and more focused queries suffer from query drift.

1. Introduction

Search engines are often promoted as the solution to the problem of information overload caused by the wealth of information made available through the World Wide Web (hereafter the web). However, search engines have a difficult time trying to sort out the relevant information from what is useless given a certain query. This is not necessarily only due to technical shortcomings with the tools but can also be attributed to human behaviour. With the expansion of the web and the widespread use of web search engines, the number of casual searchers has grown to outnumber the professional information retrieval (IR) personnel. These casual users lack both the training and the commitment associated with IR professionals and hence formulate naïve and short queries, interact with the interface in a simplistic way, and neglect to use advanced search features [17, 28]. The "vocabulary problem" [10], i.e., the ambiguity of natural languages, makes efficient retrieval difficult for short queries and hence query expansion (QE) has been developed as a solution to this problem. QE is "a process of adding new terms to a given query in an attempt to provide better contextualization (and hopefully retrieve documents which are more useful to the user)" [1: 499]. The implicit hypothesis is thus that adding more terms to the query potentially improve the effectiveness of the retrieval [27].

The problem with short queries mentioned above is not restricted to the public Web only; it is occurring also on corporate intranets. These internal webs are growing quickly; three out of every four installed web servers are intended for intranet usage [11], yet, searching in intranet settings does not receive nearly as much attention as searching on the public web. Efthimiadis [8: 146] argue, taking a user-centred approach, that to investigate the process of QE real systems, real users should be used, and our study is therefore interesting since it is carried out in a real organisational setting with real users and real documents. We apply automatic QE to a corporate intranet search engine to examine whether this would improve the quality of the results for typical (short) queries. To select expansion terms, we used Latent Semantic Indexing (LSI) [6] to find terms related to the query term initially entered by the user. Finally, we let ordinary end-users determine the relevance of the top 6 hits and then used the average document cut-off value (DCV) [12] to calculate relative precision.

The rest of the paper is organised as follows. In section two we give a short presentation of the QE field and describe our research site. Section three contains the set-up of our LSI implementation whilst section four describes the user evaluation approach. In section five we produce our results, which are subsequently discussed in section six. Section seven, finally, concludes the paper with some implications for further research.

2. Background and motivation

A substantial amount of research has been carried out in the area of query expansion. Basically three modes of QE have been identified [9]; manual, automatic, and interactive. Manual QE assumes that the user manually expands the query by adding terms and Boolean operators as part of a "building block" search strategy. Automatic QE means that terms semantically related to the query or the query terms are extracted from a thesaurus and added to the query without user intervention. Interactive QE (or semi-automatic QE), finally, typically means that possible expansion term candidates are displayed to the user who is to decide which to include in the refines query.

0-7695-2268-8/05/$20.00 (C) 2005 IEEE

Irrespective of expansion mode, QE may be based on (previously) retrieved search results or on some *a priori* knowledge structure, which in turn may or may not be collection dependent (see figure 1). Despite the amount of research carried out in this field, none of these three methods can be said to be generally superior; instead the effectiveness seems to vary greatly across settings and queries (cf. [15, 22]).

A quick analysis of the IR literature suggests that much of the research on QE is carried out using pre-arranged sets of documents such as the TREC or CLEF collections (cf. [13, 14, 17, 23, 27, 18]). The use of a common and consistent testbed in IR research is obviously useful since it allows researchers to compare the effectiveness of different techniques. However, although such sets are convenient to use, there are problems associated with such collections, especially when it comes to web searching where the reality is much more unpredictable than these testbeds. In contrast, this research was carried out at Volvo Bus Corporation (VBC), which is a manufacturing company within the Volvo Group. Volvo has a global intranet consisting of little more than 1,500 web servers, and all VBC employees have unrestricted access to this environment. Prior to the experiment described in this paper we collected log file data from Volvo's existing intranet search engine and analysed some +45,000 queries to get a broad understanding of the existing search behaviour (see table 1).

Table 1. Number of search terms used in queries submitted to Volvo's search engine

# words in queries	# queries	%	Ack. %
1 word	38,755	84.50	84.50
2 words	6,043	13.18	97.69
3 words	879	1.92	99.61
4 words	118	.26	99.87
5 words	44	.10	99.97
>5 words	17	.04	100.00
Total	**45,856**	**100.00**	**100.00**

As is evident from table 1, a large majority of the queries examined were indeed single keyword queries with an average query length of 1.18 words. This low result was expected since web searchers are known to use short queries. However, for public web searching there are data suggesting that query length might be increasing. In 1994 Pinkerton [20], examining WebCrawler, reported an average query length of 1.5 words, whereas Silverstein *et al.* [24] found the average query length for AltaVista to be 2.35 in 1998. Spink *et al.* [25] report in their 1999 paper the mean number of search terms in EXCITE to be 3.34, but that was the terms the respondents reported they *intended* to use – not actual usage. As we shall discuss later in this paper, knowing their actions are to be examined respondents may try a little harder than otherwise, which may influence the number of query terms. However, in another paper, Spink and colleagues reported the average query length to be 2.6 in 2001 [26].

Due to the unwillingness of most non-professional information seekers to invest in the additional efforts required to manually produce more relevant search queries, much of the QE research has focused on automatic methods for QE. It has been shown that for inexperienced users, interactive QE is less effective than automatic QE [15], much due to the users' inability to identify the most useful terms [22]. These results, together with our observations of non-professional information seekers in action, suggested that automatic QE would be the most feasible approach also in a typical industrial setting such as the one we were studying at Volvo.

An important difference between an intranet and the Web is that the former contains only work-related information and therefore constitutes a more homogeneous environment than does the Web. This forms a rationale for using a technique such as LSI to automatically find meaningful relationships between terms based on a collection dependent knowledge structure or a similarity thesaurus. It seems most attempts with automatic QE have used the initial user query to retrieve a small set of documents from which expansion terms are selected [2, 14, 17]. It is heuristically assumed that both the initial set of documents and the selected terms indeed are relevant but as pointed out by several commentators, these assumptions can be questioned [14, 17]. Our approach is therefore instead to follow the dotted path in figure 1 and use a pre-built thesaurus.

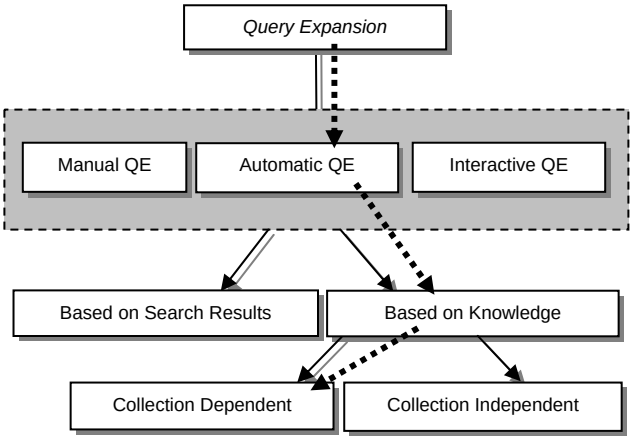

Figure 1. Possible strategies for query expansion [9: 124] and the one chosen for our experiment (dotted path)

Building and maintaining such a thesaurus is a laborious task, and in the next section we shall describe how LSI helped us automate this process.

3. Research method and setup

This research was carried out as a joint enterprise by one senior researcher and two master's level students. The senior researcher initiated and orchestrated the experiment, designed the evaluation schema, negotiated access to Volvo and supervised the entire process. The students, as part of their master thesis work, implemented the QE module, including setting up the Latent Semantic Indexing software, and assisted during the data collection phase. The data was interpreted by the senior and the students independently and the results presented in this paper come solely from the interpretation of the senior researcher.

To resolve the synonymy and polysemy problems associated with short search queries on an intranet, we wanted to build a collection dependent thesaurus. Our assumption was that a collection dependent thesaurus would not suggest synonyms in general but terms related in our specific context. If successful, this method would affect precision positively and our research hypothesis was therefore:

H0: QE will result in equal or lower precision than the unexpanded query

One approach to build automatically such a thesaurus is to use LSI to calculate co-occurrence statistics. In a vector-space model such as LSI, this is done by representing the text data as an $n \times m$ matrix where each row n represent a word and each column m represent a document. Each cell represents a normalised frequency count. Singular Value Decomposition is then applied to reduce the dimensions of the matrix by omitting all but the i largest singular values, thereby transforming the matrix to a feasible size while preserving the latent semantic relationships of the corpus words [19]. We used slightly modified (i.e., bug-fixed) version of the Telcordia Latent Semantic Indexing Software (TLSIS) (see [3] for details) for our implementation and we used i=100 dimensions, since this value had successfully been used in previous research [6, 7].

Due to limited processing and storage resources at our disposal, we could not index the entire Volvo intranet (which at the time consisted of +1,500 web servers) and had therefore to select a feasible subset. We chose Volvo Bus Corporation's subset of the intranet to be the start point, and having removed duplicates, password-protected documents, and multimedia files, we ended up with little over 1,500 VBC documents. This was too small a subset for LSI to yield reliable results, so we included yet a sub-domain, namely Volvo Group (VG). The VG domain is assumed to contain documents and information applicable to all Volvo companies. Having applied the same filtering procedure, however, we still only had 3,600 documents. We therefore added also Volvo Parts Corporation's (VPC) sub-domain to reach a total of 6,500 documents of various formats (see table 2).

Table 2. Number of documents used for the LSI process

	Volvo Bus	Volvo Parts	Volvo Group
PDF docs	325	165	423
Word docs	82	130	193
HTML docs	1126	2683	1432
Total	**1533**	**2978**	**2048**
Domain size	4405	4177	3361

Having gained access to the documents from VBC, VPC and VG, we first converted all files to HTML format using the built-in tools provided by Volvo's search engine. Once converted, we deleted all HTML tags and saved the files as plain text. To further speed up the LSI process, we also removed all stop words before feeding the result into TLSIS. From the result we created a thesaurus by first building a file with all unique terms. For each of the +47,000 terms, we then calculated the 20 closest related terms using the cosine measure. The initial term was thereafter stored in an Oracle database together with its 20 associated term/similarity value pairs sorted by descending similarity. An illustration of the database record for the term *fuel* is shown in table 3.

Table 3. The term 'fuel' and the 20 closest related terms as stored in the thesaurus

Original term, frequency	Related term, cosine measure, frequency
fuel, 1949	engine, 0.892777, 2537
	air, 0.876376, 1304
	diesel, 0.869426, 926
	exhaust, 0.866779, 853
	power, 0.856368, 711
	emission, 0.838926, 510
	oil, 0.829978, 779
	capacity, 0.798335, 497
	drive, 0.777764, 481
	low, 0.758009, 1064
	engines, 0.73928, 750
	km, 0.719719, 464
	nitrogen, 0.715228, 137
	gas, 0.713736, 659
	cleaner, 0.711149, 92
	converter, 0.702429, 425
	speed, 0.700382, 913
	temperature, 0.699393, 509
	fossil, 0,69904473, 51
	hp, 0.697893, 442

To be able to invoke our thesaurus when searching, we also built a simple web interface that would allow a user to submit a query. The query terms were intercepted by a Java program that alternately expanded the query or left it untampered with, and thereafter relayed it onto Volvo's ordinary search engine; a commercial off-the-shelf product. When receiving a "bag of words", e.g., a set of words where the order of the words is ignored, Volvo's search engine would treat such a query as an implicit OR and retrieve the documents that contained any of the words in the set. The ranking algorithm would sort documents containing many of the (significant) words to the top of the result list, using standard term frequency and inverted document frequency measures ($tf \times idf$).

When query expansion was activated, each query term provided by the user was expanded with the j most closely related thesaurus terms, provided that the similarity value exceeded 0.2, and sent to the search engine. The threshold of 0.2 was selected since it has shown to yield good results [23]. A number of initial trial runs were used to heuristically determine the number of expansion terms to j=4. During these tests we also noticed that terms with low overall frequency tended to skew the results, which is in compliance with what previously has been reported elsewhere [23]. We therefore deleted word with a frequency less than f and after some iterations we found that f=8 produced what seemed to be reliable results. This operation reduced the number of terms in our thesaurus to approximately 17,000.

4. User evaluation

The parameters most frequently used in IR systems evaluation are precision and recall, which together define a bivariate measure of retrieval effectiveness [12]. These two parameters are also interdependent; wanting to increase recall usually results in decreasing precision and vice versa [21]. Precision and recall are, however, not entirely uncontested, especially when used in web environments (cf. [5, 21]). It has been suggested that the non-professional information seekers who dominate the web are more likely to be interested in precision at top ranks than in increased recall [17], and for this reason we focus primarily on precision in this experiment. Since we wanted to examine whether search result relevance could be improved on an intranet, we based the evaluation on real user assessments.

We randomly recruited 55 users for this experiment by approaching them in their open office landscape, asking if they were willing invest half an hour in participating in a research project. Those who accepted were first asked a few short demographic questions before we briefly outlined the purpose and set-up of the experiment. We carefully explained to the users that the experiment was a test of *the system* and not of their ability to formulate good queries. In the event of a "bad" search result, we told them, this was to be understood as the system's fault and not as an indication of them lacking searching competence. We never explained to the users that query expansion was being tested; we simply told them that two different approaches were being evaluated.

Having spent some time browsing through the VBC domain, familiarising ourselves with the content, we had arbitrary constructed five search tasks that we wanted our test users to complete. The tasks were formed from data on actual intranet pages, ensuring that they could be answered, and were constructed to simulate a real life search situation (see figure 2 for an illustration of task #3). At this point we did not consider whether the queries where targeted or explorative, something we shall discuss in more detail in section 6.

Figure 2. Simulated search task

Task #3.

Exhaust emission from vehicles is a contributory cause to the greenhouse effect. What measures does Volvo take to reduce emissions?

For each of the five search tasks, the users were told to formulate a query as they would normally do and submit it using our interface as described earlier. Only if the user typed a syntactically illegal query or made an obvious typing error, did we interfere by correcting the query using the same term(s). The top six results from the internal search engine were retrieved and evaluated by the user by stating for each result whether or not it was considered relevant. When all six entries had been evaluated we rerun the same query using the alternative approach, and the users were again asked to evaluate the result. Often, the two approaches returned partly overlapping result sets and when that happened, we consistently used the evaluation given from the first occasion. Approximately half of the users were first presented with the expanded results and then the unexpanded results while the other half had it in reversed order. At runtime neither the user nor the researcher was able to determine which result set came from which algorithm. In total, we spent 30-50 minutes with each user.

Afterwards, the results from the 55 users were used to calculate the relative precision using the document cut-off value (DCV). By holding the number of retrieved documents constant at 6 hits, we calculated precision for each query relying on the users' opinion as to whether or not a particular document was relevant. We chose to set the DCV fairly low for three reasons; i) users are most interested in top ranked documents [17]; ii) high DCV values causes precision to deteriorate [12], and; iii) we

assumed it to be difficult to motivate users to evaluate a large number of documents per query.

To minimise the risk of receiving biased results we followed Hull's [12] advice and calculated precision over a range of DCV's (1 to 6 in our case) and averaged the results, as in formula 1.

$$\text{Average precision} = 1/6 \sum_{i=1}^{6} \text{precision}(DCV_i) \qquad (1)$$

Such an approach also favours situations were the relevant documents are high up in the result set, which most casual information seekers intuitively seem to appreciate.

5. Results

With no more than 17,000 terms in our thesaurus, there was an obvious risk that the users would use query terms that were not in our thesaurus and therefore could not be expanded. This also turned out to be the true in 67 of the 275 cases (~25%). When a query could not be expanded, the user de facto received the result set from an unexpanded query, i.e., the two approaches returned identical results. These queries are marked with an asterisk in the table in Appendix A and the corresponding pair was disregarded when calculating the final result. Due to this correction, we ended up with 208 pairs.

A two-tailed paired *t* test was used to determine whether or not the differences between the unexpanded and the expanded queries were significant. To our disappointment the result seemed to favour the *un*-expanded queries but the difference between the two averaged precision measures was small (0.464 for Unexpanded vs. 0.451 for Expanded) and not statistically significant. Therefore we could not reject the null hypothesis and we were not able to conclude that automatic QE using a collection dependent knowledge structure on an intranet generally would increase precision. See table 4 for details.

Table 4. Overall difference between unexpanded and expanded queries

	Unexpanded	Expanded
Mean	0.464	0.451
Variance	0.116	0.154
N	208	208
df		207
| t |		0,620
Critical values for two-tailed *t* distribution,		
	5% level	1.97
	1% level	2.60

However, there were distinct differences between individual queries as is evident from Appendix A, and when analysing the queries individually we fount most of these differences were also statistically significant. Query #1 (Q1) showed a significantly better result for the unexpanded query (see table 5).

Table 5. Differences between unexpanded and expanded results for query #1

	Unexpanded	Expanded
Mean	0.390	0.161
Variance	0.0624	0.0383
N	40	40
df		39
| t |		5.5746
Critical values for two-tailed *t* distribution,		
		.05 level 2.023
		.01 level 2.708
		.001 level 3.560

Query #2 (Q2), in contrast, showed a better result for the expanded query and also this difference was significant, albeit not as evident as for Q1. See the details in table 6.

Table 6. Differences between unexpanded and expanded results for query #2

	Unexpanded	Expanded
Mean	0.534	0.590
Variance	0.120	0.158
N	50	50
df		49
| t |		2.324
Critical values for two-tailed *t* distribution,		
		.05 level 2.010
		.01 level 2.680

As for Q2, Query #3 also performed better when expanded but Q3 had more drop-outs (occasions when the query term could not be expanded) than had Q1 and Q2 and difference between the two approaches was not statistically significant, as can be seen from table 7.

Table 7. Differences between unexpanded and expanded results for query #3

	Unexpanded	Expanded
Mean	0.595	0.671
Variance	0.128	0.112
N	35	35
df		34
| t |		1.700
Critical values for two-tailed *t* distribution,		
		.05 level 2.032
		.01 level 2.728

Also Query #4 had many drop-outs but the difference between the two approaches was still significant. For Q4 the unexpanded queries again produced the better result (see table 8).

Table 8. Differences between unexpanded and expanded results for query #4

	Unexpanded	Expanded
Mean	0.235	0.109
Variance	0.0983	0.0556
N	34	34
df		33
| t |		3.0819
Critical values for two-tailed *t* distribution,		
	.05 level	2.035
	.01 level	2.733

Query #5, finally, did again favour the expanded approach and also this time was the difference significant. Refer to table 9 for details.

Table 9. Differences between unexpanded and expanded results for query #5

	Unexpanded	Expanded
Mean	0.517	0.627
Variance	0.107	0.112
N	49	49
df		48
| t |		2.120
Critical values for two-tailed *t* distribution,		
	.05 level	2.011
	.01 level	2.682

Observing that Queries #1 and #4 obviously performed better when not being expanded whilst Queries #2, #3, and #5 seemed to benefit from expansion, we also tested the results from the pooled queries. Taken together, Q1 and Q4 produced a significantly better result when unexpanded, as is shown in table 10.

Table 10. Differences between unexpanded and expanded results for pooled queries #1 and 4.

	Unexpanded	Expanded
Mean	0.319	0.137
Variance	0.0838	0.0463
N	74	74
df		73
| t |		6.158
Critical values for two-tailed *t* distribution,		
	.05 level	1.993
	.01 level	2.644
	.001 level	3.440

Also Q2, Q3, and Q5 taken together produced a significant difference between the two approaches and as expected it was in favour of the expanded approach (see table 11).

Table 11. Differences between unexpanded and expanded results for pooled queries #2, 3 and 5

	Unexpanded	Expanded
Mean	0.544	0.624
Variance	0.117	0.129
N	134	134
df		133
| t |		3.381
Critical values for two-tailed *t* distribution,		
	.05 level	1.99
	.01 level	2.63
	.001 level	3.39

To summarise, we were testing whether query expansion would have an effect on search precision as experienced by end users, but were not able to draw a general conclusion on the basis of our results. However, we did notice significant differences between the two approaches when testing queries individually and also when pooling queries (see table 12).

Table 12. Significance levels for differences between expanded and unexpanded queries, individually and clustered

Q1	$p = 0.000$**
Q2	$p = 0.025$*
Q3	$p = 0.099$
Q4	$p = 0.005$**
Q5	$p = 0.040$*
Q1+4	$p = 0.000$**
Q2+3+5	$p = 0.001$**

In the following section, we shall interpret and discuss the results just presented and suggest a tentative rationale for the query pooling that was used.

6. Discussion

Although this study was conducted in a quantitative way with statistical analysis to test the significance of the results, it was in a sense an explorative study. We hypothesed that QE would have a positive impact but we had no theory of exactly why or to what extent. When constructing our test tasks we took departure from actual intranet pages found when surfing the VBC intranet. This way we knew that our questions would have (at least) one answer. However, noticing the result and the clear difference between the Q1+Q4 cluster and the

Q2+Q3+Q5 cluster we in retrospect analysed the nature of the tasks more carefully. We posit that Q2, Q3 and Q5 differs in nature from Q1 and Q4 inasmuch as they are not necessarily answered by one exhaustive statement but by a set of partial answers that together give an holistic picture. We shall refer to such queries as *explorative* questions. Task #2 (see figure 3 below) and task #3 (figure 2 earlier) illustrate such explorative tasks where the users could search extendedly without finding an exhaustive answer. The answer to an explorative question is typically distributed across several documents from several authors due to the fact that no single person is likely to know all the facts. The seeker must therefore collect different pieces from different sources.

Task #2.

On what technology does Volvo base its strategy for alternative fuel?

Figure 3. An explorative query task

Q1 and Q4, in contrast, could be expected to have rather precise (albeit not necessarily unique) answers. We shall call these queries *targeted* questions and task #4 (see figure 4) is an illustration of such a query. The answer to task #4 might be a function within VBC (e.g., your personnel administrator), the name of the person occupying that position (e.g., Maria Ericsson), or possibly a department where the function is hosted (e.g., the HR department), and any one of these answers in isolation would have been sufficient. In the case of a targeted query there is no (or less) need for the seeker to collect and synthesise many documents to find the answer.

Task #4.

You are purchasing a Volvo automobile for private use. Who at VBC should you contact to receive your corporate discount voucher?

Figure 4. A targeted query task

A documented weakness with automatic QE is the obvious risk of query drift, i.e., "the alteration of the focus of a search topic caused by improper expansion" [17: 206]. We posit that this drift is what caused the negative results for the targeted queries. However, an interesting and novel result of this study is that query drift should perhaps not always be interpreted as "improper" expansion but rather as *unexpected* expansion. Although our results suggest that query drift may indeed be hurting precision in the case of a targeted query, just as may be expected, the broadening of the search scope that the query expansion results in seems to improve precision for explorative queries. This could very well be due to query drift, and query drift may therefore not be entirely bad.

We have argued that organisational members seeking information on their intranet are seldom trained IR professionals but more casual seekers. It would therefore be wrong, we suggest, describing their actions in terms of information retrieval. A more accurate description would be information seeking, which connotes a more open-ended activity and is defined as "a process in which people humans purposefully engage in order to change their state of knowledge" [16: 5]. Research has confirmed that organisational members use undirected browsing or conditioned viewing as their principal strategies to satisfy their information needs [4]. It is therefore plausible that organisational members use explorative searching more frequently than targeted searching, and hence would benefit from QE. This is an interesting hypothesis that could be tested in future research.

The research described in this paper is limited in scope (e.g. only five queries) and was not designed from the outset to test this query drift theory, so the results need to be verified with more research. Nevertheless, we believe the results show that there is a difference between explorative and targeted search patterns and that this should be considered when designing future QE systems, particularly so for intranets. Not only have different users different search patterns, but the same individual may alter between different modes and this presents a challenge to QE system designers.

Our study also has other limitations. It may be argued that a collection of 6,500 documents is still too small for LSI to work properly. This limitation was forced by the lack of hardware resources and the fact that LSI requires all documents to be collected prior to processing. Although we found LSI to be useful for the purpose of QE and for automatically build a thesaurus we also note that it is difficult and resource consuming to update the thesaurus regularly. An algorithm similar to LSI but with better scalability is Random Indexing [23] and it may be a better idea to base subsequent experiments on this algorithm.

It might also have been better to use a document frequency threshold rather than a term frequency threshold. We now removed all terms with an overall frequency less than f=8. However, a number of terms were used frequently in just one or a few documents but nowhere else and this might have created strange relationships. It might have been a good idea to also delete terms that occurred in less than d documents. The best value of d has to be empirically derived.

Finally, there is the bias that comes from the users knowing that they participated in an experiment. Although we explained that we were not testing the users' ability to search, it is evident that their ambition to "do well" affected their search behaviour. For example, the average query length in our test was 2.18 words as opposed to the 1.18 seen when analysing search logs.

Note, though, that in both cases are our numbers lower than those reported by Spinks and Silverstein. This may be due to the fact that most Volvo Bus employees used Swedish query terms. The Swedish language uses compound words, whereas e.g. English uses multiple words. The English 2-word query "diesel engine" would in Swedish translate to the single word query "dieselmotor". It seems plausible that this characteristic had an impact on the query length and a follow-up study of an English intranet is therefore underway to determine whether this result was indeed language related or if intranet queries are shorter.

7. Conclusions

Using LSI, we have built a collection dependent similarity thesaurus, which has been used to expand search engine queries on a corporate intranet. Given the specifics of an intranet, we were hoping that such an approach would produce increased search result quality but the mixed results received did not support this hypothesis.

Instead, our post hoc analysis suggests that we have been using two different categories of questions; *explorative* and *targeted*. Our conclusion is that whilst targeted queries seem to suffer from the query drift that automatic QE may produce, explorative queries appear to benefit from such side-effects. This suggests, we claim, that QE may be appropriate for information seeking rather than for information retrieval. Since most organisational members are non-IR professionals, and hence more likely to engage in information seeking, QE may be a useful technique on corporate intranets. We believe this to be a useful new insight that has important implications for future QE research.

A secondary result is the observation that query drift *per se* should not always be considered altogether bad. The improper expansion previously associated with query drift could instead be interpreted as unexpected expansion, where the latter may lead to hits otherwise never found. This sort of expansion seems to be particularly useful for explorative search.

8. Acknowledgement

The author wishes to thank Volvo Bus Corporation for allowing us access to their intranet and volunteering to participate in this study. Thanks are also due to Karin R:dotter Mark and Cecilia Koskinen for their thorough and dedicated work with the QE prototype, the thesaurus, and the user evaluations.

9. References

[1] Baeza-Yates, R. and Ribiero-Neto, B., *Modern Information Retrieval*, Addison-Wesley, 1999.

[2] Billerbeck, B. and Zobel, J., "When Query Expansion Fails", in Proceedings of SIGIR '03, ACM Press, Toronto, Canada, 2003, pp. 387-388.

[3] Chen, C., Stoffel, N., Post, M., Basu, C., Bassu, D., and Behrens, C., "Telcordia LSI Engine: Implementation and Scalability Issues", in Proceedings of 11th Int'l Workshop on Research Issues in Data Engineering, Heidelberg, Germany, 2001.

[4] Choo, C. W., Detlor, B., and Turnbull, D., *Web Work: Information Seeking and Knowledge Work on the World Wide Web*, Kluwer Academic Press, 2000.

[5] Chu, H. and Rosenthal, M., "Search Engine for the World Wide Web: A Comparative Study and Evaluation Methodology", in Proceedings of ASIS '96, Baltimore, MD, 1996, pp. 127-135.

[6] Deerwester, S., Dumais, S., Furnas, G., Landauer, T., and Harshman, R., "Indexing by latent semantic analysis", *Journal of the American Society for Information Science*, 41(6), 1990, pp. 391-407.

[7] Dumais, S., Landauer, T., and Littman, M.L., "Automatic Cross-Language Information Retrieval using Latent Semantic Indexing", in Proceedings of SIGIR '96 Workshop on Cross-Linguistic Information Retrieval, 1996, pp. 16-23.

[8] Efthimiadis, E. "A user-centred evaluation of ranking algorithms for interactive query expansion", In Proceedings of SIGIR '93, Pittsburgh, PA, pp. 146-159.

[9] Efthimiadis, E., "Query Expansion", in M. E. Williams (ed.) *Annual Review of Information Science and Technology*, 31, 1996, pp. 121-187.

[10] Furnas, G., Landauer, T., Gomez, L., and Dumais, S., "The Vocabulary Problem in Human-Systems", *Communication, Communications of the ACM*, 30(11), 1987, pp. 964-971.

[11] Gerstner, J., "Intranets mean business", *Communication World*, 19(2), Feb./March 2002, pp 14-17.

[12] Hull, D., "Using Statistical Testing in the Evaluation of Retrieval Experiments", in Proceedings of SIGIR '93, ACM Press, Pittsburgh, PA, 1993, pp. 329-338.

[13] Kang, I. and Kim, G., "Query Type Classification for Web Document Retrieval", in Proceedings of SIGIR '03, ACM Press, Toronto, Canada, 2003, pp. 64-71.

[14] Mano H. and Ogawa, Y., "Selecting Expansion Terms in Automatic Query Expansion", in Proceedings of SIGIR '01, ACM Press, New Orleans, LA, 2001, pp. 390-391.

[15] Magennis, M. and van Rijsbergen, C.J., "The potential and actual effectiveness of interactive query expansion", In Proceedings of SIGIR '97, Philadelphia, PA., pp. 324-332.

[16] Marchionini, G., *Information Seeking in Electronic Environments*, Cambridge University Press, 1995.

[17] Mitra, M., Singhal, A., and Buckley, C., "Improving Automated Query Expansion", in Proceedings of SIGIR '98, ACM Press: Melbourne, Australia, 1998, pp. 206-214.

[18] Ogilvie, P. and Callan, J., "The Effectiveness of Query Expansion for Distributed Information Retrieval", In Proceedings of CIKM '01, Atlanta, GA., pp. 183-190.

[19] Papadimitriou, C., Raghavan, P., Tamaki, H., and Vempala, S., "Latent Semantic Indexing: A probabilistic analysis", *Journal of Computer and System Sciences*, 61(2), 2000, pp 217-235.

[20] Pinkerton, B., "Finding What People Want: Experiences with the WebCrawler", in Proceedings of WWW '94, Chicago, IL., 1994.

[21] Raghavan, V., Bollman, P., and Jung, G.S., "A critical investigation of recall and precision as measures of retrieval system performance", *Communication of the ACM*, 7(3), 1989, pp. 205-229.

[22] Ruthven, I., "Re-examining the Potential Effectiveness of Interactive Query Expansion", In Proceedings of SIGIR '03, Toronto, Canada, pp. 213-220.

[23] Sahlgren, M., Karlsgren, J., Cöster, R., and Järvinen, T., "Automatic Query Expansion using Random Indexing", in Proceedings of CLEF 2002, Rome, Italy.

[24] Silverstein, C., Henzinger, M., Marais, H., and Moricz, M., "Analysis of a very large AltaVista query log", SRC Technical Note 1998-14, October 26, Digital Equipment Corp., 1998.

[25] Spink, A., Bateman, J., and Jansen, M. B. J., "Searching the Web: A survey of EXCITE users", *Internet Research*, 9(2), 1999, pp. 117-128.

[26] Spink, A., Wolfram, D., Jansen, M. B. J., and Saracevic, T., "From e-sex to e-commerce: Web search changes", *IEEE Computer*, 35(3), 2002, pp. 107-109.

[27] Wei, J., Qui, Z., Bressan, S., and Ooi, B. C., "Mining Term Association Rules for Global Query Expansion: A Case Study with Topic 202 from TREC4", In Proceedings of AMCIS 2000, pp. 85-90.

[28] White, R., Ruthven, I., and Jose, J., "Finding Relevant Documents using Top Ranking Sentences: An Evaluation of Two Alternative Schemes", In Proceedings of SIGIR '02, ACM Press, Tampere, Finland, 2002, pp. 57-64.

0-7695-2268-8/05/$20.00 (C) 2005 IEEE

Appendix A. Average relative precision for the five queries from 55 test users. The +e denotes an expanded query.

n	Q1	Q1+e	Q2	Q2+e	Q3	Q3+e	Q4	Q4+e	Q5	Q5+e
1	.408	0	0	0	.611	*	.158	0	0	.242
2	.939	*	.808	.808	0	*	0	*	0	.433
3	.750	*	0	0	0	0	0	0	1.000	.917
4	.158	.242	0	0	.192	0	.103	0	.808	*
5	.408	.242	.897	1.000	.242	1.000	.242	0	.408	.103
6	.408	.242	0	0	0	*	0	0	.219	.408
7	.242	*	.808	.808	.833	*	.436	*	.836	.911
8	.408	0	.808	.836	1.000	1.000	.567	*	.836	.911
9	.650	0	.408	0	0	0	.408	*	.269	.517
10	.436	.428	.650	.650	.808	.808	0	*	.836	.572
11	.503	.428	.808	.808	1.000	.842	0	*	.808	.408
12	.869	0	.808	.808	.261	.650	0	0	.242	.711
13	.158	*	0	0	1.000	*	.306	.306	.061	1.000
14	.469	0	.808	.808	.158	*	0	0	.089	0
15	.436	*	.808	.869	0	*	.808	*	.808	.911
16	0	.028	.567	.808	.911	.808	0	*	.808	.650
17	.511	*	0	0	1.000	*	0	*	.753	.808
18	0	*	.103	*	.408	.567	.158	.158	.028	.739
19	.722	.722	1.000	1.000	.972	.650	0	0	.372	0
20	.870	*	.808	.808	.854	*	.436	*	.408	1.000
21	.469	0	.808	.808	.322	.808	.650	.103	.842	.842
22	0	0	1.000	1.000	1.000	1.000	.869	0	.678	.089
23	0	0	0	*	0	*	.061	*	1.000	*
24	0	0	.697	.972	1.000	.808	0	0	.592	1.000
25	.408	.242	.408	1.000	.628	.869	1.000	.972	.869	1.000
26	.408	.242	.408	.408	0	*	0	*	0	*
27	.408	.242	.408	.408	.650	.678	0	*	1.000	*
28	0	*	.842	1.000	.408	.567	.158	.158	.028	.739
29	.722	.722	1.000	1.000	.972	.650	0	0	.372	0
30	.870	.242	.808	.808	.972	1.000	.269	*	.408	1.000
31	0	.028	.567	.808	.911	.808	.436	.103	.808	.650
32	.650	0	.408	0	0	0	.408	.103	.269	.517
33	0	0	0	*	0	*	.061	*	1.000	*
34	.869	*	.753	.808	0	*	0	0	0	.433
35	.750	*	0	0	0	0	0	0	1.000	.917
36	.408	.242	.372	.408	0	*	0	*	0	*
37	.408	.242	.408	.433	.650	.678	0	*	.650	1.000
38	.469	0	.808	1.000	.322	.808	.592	.103	.842	.842
39	0	0	.911	1.000	.854	*	.869	0	.678	.089
40	.408	.242	.103	*	.242	1.000	.242	.242	.408	.103
41	0	0	.697	.972	.808	.808	0	0	.592	1.000
42	.408	.242	.408	1.000	.628	.869	1.000	.972	.869	1.000
43	.158	.242	0	0	.192	0	.103	0	.592	.711
44	.869	0	.753	.808	.261	.650	0	0	.242	.711
45	.408	.242	0	0	0	*	0	0	.219	.408
46	.242	*	.433	.808	.833	1.000	.219	*	.836	.911
47	.436	.408	.592	.650	.650	.808	0	.044	.836	.572
48	.503	.428	.808	.808	1.000	.842	0	0	.808	.408
49	.433	0	0	0	.433	*	.158	.044	0	.242
50	.469	0	.808	.808	.158	*	0	0	.089	0
51	.511	*	0	0	1.000	*	0	*	.753	.808
52	.436	*	.808	.869	0	*	.808	*	.808	.911
53	.158	*	0	.044	1.000	.842	.306	.306	.061	1.000
54	.408	0	.808	.836	1.000	1.000	.567	*	.836	.911
55	.436	.103	.433	*	.572	.650	.219	.103	.572	.650
	.409	**.161**	**.497**	**.590**	**.504**	**.671**	**.229**	**.109**	**.530**	**.627**