Mining Context Specific Similarity Relationships Using The World Wide Web

Dmitri Roussinov
Department of Information Systems
W.P. Carey School of Business
Arizona State University
Tempe, AZ, 85287
dmitri.roussinov@asu.edu

Leon J. Zhao
Department of Management Information Systems
University of Arizona
Tucson, AZ 85721
lzhao@bpa.arizona.edu

Weiguo Fan
Department of Information Systems
Virginia Tech
Blacksburg, VA 24061
wfan@vt.edu

Abstract

We have studied how context specific web corpus can be automatically created and mined for discovering semantic similarity relationships between terms (words or phrases) from a given collection of documents (*target collection*). These relationships between terms can be used to adjust the standard vectors space representation so as to improve the accuracy of similarity computation between text documents in the target collection. Our experiments with a standard test collection (Reuters) have revealed the reduction of similarity errors by up to 50%, twice as much as the improvement by using other known techniques.

1 Introduction

Many modern information management tasks such as document retrieval, clustering, filtering and summarization rely on algorithms that compute similarity between text documents. For example, clustering algorithms, by definition, place documents similar to each other into the same cluster. Topic detection algorithms attempt to detect documents or passages similar to those already presented to the users. "Query by example" retrieval is based on similarity between a document selected as example and the other ones in the collection. Even a classical retrieval task can be formulated as rank ordering according to the similarity between the document (typically very short) representing user's query and all the documents in the collection.

For similarity computation, text documents are represented by *terms* (words or phrases) that they have, and encoded by vectors according to a predominantly used vector space model (Salton & McGill, 1983). Each coordinate corresponds to a *term* (word or phrase) possibly present within a document. Within that model, a high similarity between a pair of documents can be only indicated by sharing same terms. This approach has apparent limitations due to the notorious vocabulary problem (Furnas et al., 1997): people very often use different words to describe semantically similar objects. For example, within a classical vector space model, the similarity algorithm would treat words *car* and *automobile* as entirely different, ignoring semantic similarity relationship between them.

It has been known for a long time that semantic similarity relationships between terms can be discovered by their co-occurrence in the same documents or in the vicinity of each other within documents (von Rijsbergen, 1977). Until the 1990s, the studies exploring co-occurrence information for building a thesaurus and using it in automated query expansion (adding similar words to the user query) resulted in mixed results (Minker et al., 1972; Peat & Willett, 1991). The earlier difficulties may have resulted from the following reasons:

1) The test collections were small, sometimes only few dozens of documents. Thus, there was only a small amount of data available for statistical co-occurrence analysis (mining), not enough to establish reliable associations.

2) The evaluation experiments were based on retrieval tasks, short, manually composed queries. The queries were at times ambiguous and, as a result, wrong terms were frequently added to the query. E.g. initial query "jaguar" may be expanded with the words "auto", "power", "engine" since they co-occur with "jaguar" in auto related documents. But, if the user was actually referring to an animal then the retrieval accuracy would degrade after the expansion.

3) The expansion models were overly simplistic, e.g. by merely adding more keywords to Boolean queries (e.g. "jaguar OR auto OR power OR car").

Although more recent works removed some of the limitations and produced more encouraging results (Grefenstette, 1994; Church et al., 1991; Hearst et al., 1992; Schutze and Pedersen, 1997; Voorhees, 1994) there are still a number of questions that remain open:

1) What is the range for the magnitude of the improvement. Can the effect be of practical importance?

2) What are the best mining algorithms and formulas? How crucial is the right choice of them?

3) What is the best way to select a corpus for mining? Specifically, is it enough to mine only within the same collection that is involved in retrieval, clustering or other processing (*target collection*), or constructing and mining a larger

Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP), pages 499–506, Vancouver, October 2005.

external corpus (like a subset of World Wide Web) would be of much greater help?
4) Even if the techniques studied earlier are effective (or not) for query expansion within the document retrieval paradigm, are they also effective for a more general task of document similarity computation? Similarity computation stays behind almost all information retrieval tasks including text document retrieval, summarization, clustering, categorization, query by example etc. Since documents are typically longer than user composed queries, their vector space representations are much richer and thus expanding them may be more reliable due to implicit disambiguation.
Answering these questions constitutes the novelty of our work. We have developed a Context Specific Similarity Expansion (CSSE) technique based on word co-occurrence analysis within pages automatically harvested from the WWW (Web corpus) and performed extensive testing with a well known Reuters collection (Lewis, 1997). To test the similarity computation accuracy, we designed a simple combinatorial metric which reflects how accurately (as compared to human judgments) the algorithm, given a document in the collection, orders all the other documents in the collection by the perceived (computed) similarity. We believe that using this metric is more objective and reliable than trying to include all the traditional metrics specific to each application (e.g. recall/precision for document retrieval, type I/II errors for categorization, clustering accuracy etc.) since the latter may depend on the other algorithmic and implementation details in the system. For example, most clustering algorithms rely on the notion of similarity between text documents, but each algorithm (k-means, minimum variance, single link, etc.) follows its own strategy to maximize similarity within a cluster.
We have found out that our CSSE technique have reduced similarity errors by up to 50%, twice as much as the improvement due to using other known techniques such as Latent Semantic Indexing (LSI) and Pseudo Relevance Feedback (PRF) within the same experimental framework. In addition to this dramatic improvement, we have established the importance of the following for the success of the expansion: 1) using external corpus (a constructed subset of WWW) in addition to the target collection 2) taking the context of the target collection into consideration 3) using the appropriate mining formulas. We suggest that these three crucial components within our technique make it significantly distinct from those explored early and also explain more encouraging results.
The paper is structured as follows. Section 2 discusses previous research results that are closely related to our investigation. Section 3 presents algorithms implemented in our experiments. Section 4 describes our experiments including error reduction, sensitivity analysis, and comparison with other techniques. Finally, Section 5 concludes the paper by explaining our key contributions and outlining our future research.

2 Related Work

Most of the prior works performed only mining within the target collection itself and revealed results ranging from small improvements to negative effects (degrading performance). Throughout our paper, we refer to them as *self-mining* to distinguish from mining *external* corpus, which we believe is more promising for similarity computation between documents due to the following intuitive consideration. Within self-mining paradigm, terms *t1* and *t2* have to frequently co-occur in the collection in order to be detected as associated (synonymic). In that case, expanding document *D* representation with a term *t2* when the document already has term *t1* is not statistically likely to enrich its representation since t2 is likely to be in document *D* anyway. We believe mining external larger and contextually related corpus has the potential to discover more interesting associations with much higher reliability than just from the target collection. That is why, this paper focuses on constructing and mining the external corpus.
There are very few studies that used external corpus and standard evaluation collections. Grefenstette (1994) automatically built a thesaurus and applied it for query expansion, producing better results than using the original queries. Gauch et al. (1998) used one standard collection for mining (TREC4) and another (TREC5) for testing and achieved 7.6% improvement. They also achieved 28.5% improvement on the narrow-domain Cystic Fibrosis collection. Kwok (1998) also reported similar results with TREC non Web collections. Ballesteros and Croft (1998) used unlinked corpora to reduce the ambiguity associated with phrasal and term translation in Cross-Language Retrieval.
There are even fewer studies involving semantic mining on the Web and its methodological evaluation. Géry and Haddad Géry (1999) used about 60,000 documents from one specific domain for mining similarity among French terms and tested the results using 4 ad hoc queries. Sugiura and Etzioni (2000) developed a tool called Q-Pilot that mined the web pages retrieved by commercial search engines and expanded the user query by adding similar terms. They reported preliminary yet encouraging results but tested only the overall system, which includes the other, not directly related to mining features, such as clustering, pseudo-relevance feedback, and selecting the appropriate external search engine. Furthermore, they only used the correctness of the engine selection as the evaluation metric. There are some other well known techniques that do not perform mining for a thesaurus explicitly but still capture and utilize semantic similarity between the terms in an implicit way, namely Latent Semantic Indexing (LSI) and Pseudo Relevance Feedback (PRF). Latent Semantic Indexing (Analysis) (Deerwester et al., 1998) a technique based on Singular Value Decomposition, was studied in a number of works. It reduces the number of dimensions in the document space thus reducing the noise (linguistic variations) and bringing semantically similar terms together, thus it

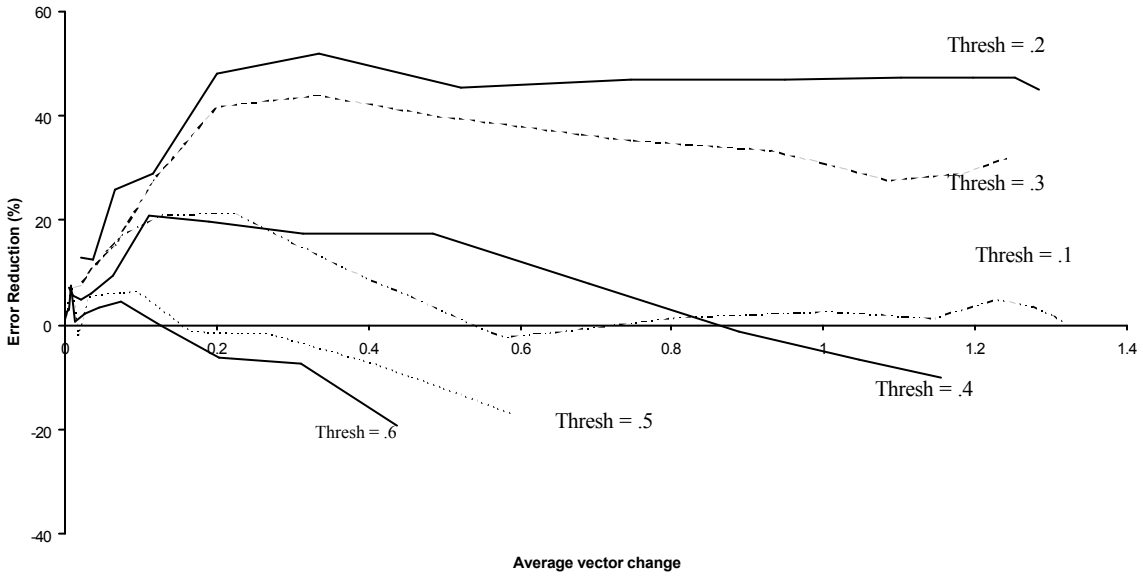

Figure 1. The average error reduction (%) as a function of average document vector change *Ca* for various threshold parameters *Thresh*.

takes into consideration the correlation between the terms. The reported improvements so far however have not exceeded 10-15% in standard collections) and sensitive to the choice of the semantic axis (reduced dimensions). The general idea behind the Pseudo Relevance Feedback (PRF) (Croft & Harper, 1979) or its more recent variation called Local Context Analysis (Xu & Croft, 2000) is to assume that the top rank retrieved documents are relevant and use certain terms from them for the query expansion. A simple approach has been found to increase performance over 23% on the TREC3 and TREC4 collections and became internal part of modern IR systems. Although this idea has been only applied so far to users' queries, we extended it in this study to similarity computation between documents in order to compare with our approach. Although we believe this extension is novel, it is not the focus of this study. It is also worth mentioning that both LSI and PRF fall into "self-mining" category since they do not require external corpus.

A manually built and maintained ontology (a thesaurus), such as WorldNet, may serve as a source of similarity between terms and has been shown to be useful for retrieval tasks (Voorhees, 1994). However, one major drawback of manual approach is high cost of creating and maintaining. Besides, the similarity between terms is context specific. For example, for a campus computer support center the words *student*, *faculty*, *user* are almost synonyms, but for designers of educational software (e.g. Blackboard), the words *student* and *faculty* would represent entirely different roles.

Although the terms "mining", "web mining" and "knowledge discovery" have been used by other researchers in various contexts (Cooley, 1997), we believe it is legitimate to use them to describe our work for two major reasons: 1) We use algorithms and formulas coming from the data mining field, specifically signal to noise ratio association metric (Church, 1989; Church, 1991) 2) Our approach interacts with commercial search engines and harvests web pages contextually close to the target collection, and there is mining of resources (the search engine database) and discovery of content (web pages) involved. We admit that the term "mining" may be also used for a more sophisticated or different kind of processing than our approach here.

3 Algorithms And Implementations

The target collection (Reuters in our experiment) is indexed and its most representative terms are used to construct a corpus from an external source (e. g. World Wide Web). The term-to-term similarity matrix is created by co-occurrence analysis within the corpus and subsequently used to expand document vectors in order to improve the accuracy (correctness) of similarity computation between the documents in the target collection. Although in this work we do not study the effects on the individual applications of the similarity computation, it is crucial for such tasks as retrieval, clustering, categorization or topic detection.

3.1 Building a Web Corpus

We designed and implemented a heuristic algorithm that takes advantage of the capabilities provided by commercial web search engines. In our study, we used AltaVista (www.altavista.com), but most other search engines would also qualify for the task. Ideally, we would like to obtain web pages that

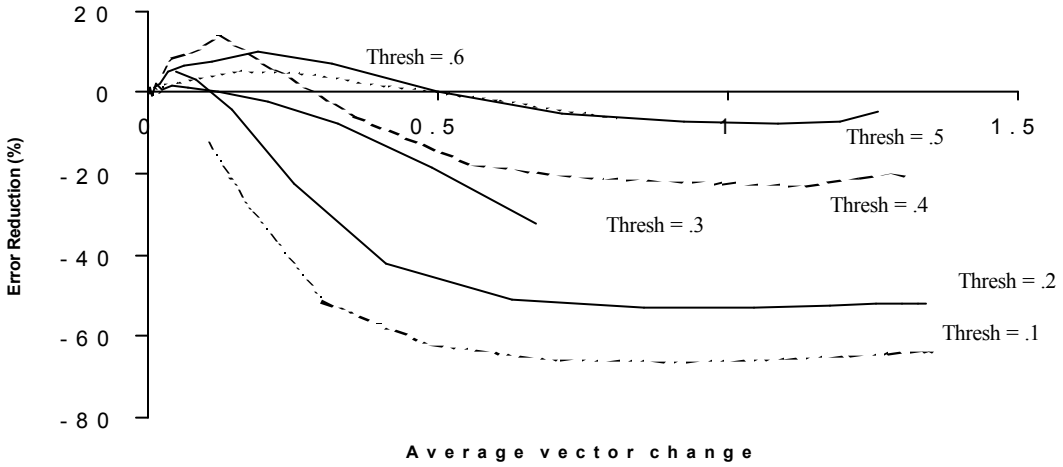

Figure 2. The average error reduction (%) as a function of average document vector change *Ca* for various threshold parameters *Tresh* without "context hint" terms.

contain the terms from the target collection in the similar context. While constructing Web corpus, our spider automatically sends a set of queries to AltaVista and obtains the resulting URLs. The spider creates one query for each term t_i out of 1000 most frequent terms in the target collection (stop words excluded) according to the following formula:

$$q_i = \text{"+"} + t_i + \text{" "} + context\_hint$$

, where + means string concatenation, quotes are used to represent text strings literally and *context_hint* is composed of the top most frequent terms in the target collection (stop words excluded) separated by empty space. Although this way of defining context may seem a bit simplistic, it still worked surprisingly well for our purpose.

According to AltaVista, a word or phrase preceded by '+' sign has to be present in the search results. The presence of the other words and phrases (context hint string in our case) is only desirable but not required. The total number of the context hint terms (108 in this study) is limited by the maximum length of the query string that the search engine can accept.

We chose to use only top 1000 terms for constructing corpus to keep the downloading time manageable. We believe using a larger corpus would demonstrate even larger improvement. Approximately 10% of those terms were phrases. We only used the top 200 hits from each query and only first 20Kbytes of HTML source from each page to convert it into plain text. After removing duplicate URLs and empty pages, we had 19,198 pages in the Web corpus to mine.

Downloading took approximately 6 hours and was performed in parallel, spawning up to 20 java processes at a time, but it still remained the largest scalability bottleneck.

3.2 Semantic Similarity Discovery

CSSE performs co-occurrence analysis at the document level and computes the following values: *df(t1, t2)* is the joint document frequency, i.e., the number of web pages where both terms *t1* and *t2* occur. df(t) is the document frequency of the term t, i.e., the number of web pages in which the term t occurs. Then, CSSE applies a well known signal to noise ratio formula coming from data mining (Church, 1991) to establish similarity between terms *t1* and *t2*:

$$sim(t1, t2) = \log \frac{N \cdot df(t1,t2)}{df(t1) \cdot df(t2)} / \log N, \qquad (1)$$

where N is the total number of documents in the mining collection (corpus),

$log\ N$ is the normalizing factor, so the *sim* value would not exceed *1* and be comparable across collections of different size.

Based on the suggestions from the other studies using formula (1), before running our tests, we decided to discard as spurious all the co-occurrences that happened only within one or two pages and all the similarities that are less than the specified threshold *(Thresh)*.

3.3 Vector Expansion

Since we were modifying document vectors (more general case), but not queries as in the majority of prior studies, we refer to the process as *vector expansion*. As we wrote in literature review, there are many possible heuristic ways to perform vector expansion. After preliminary tests, we settled on the simple linear modification with post re-normalization as presented below. The context of the target collection is represented by the similarity matrix *sim(t1, t2)* mined as described in the preceding section. Our vector expansion algorithm adds all the related terms to the vector representation of the document *D* with the weights proportional to the degree of the relationships and the global inverse document frequency (IDF) weighting of the added terms:

$w(t, D)' = w(t, D) +$

$$a \sum_{t1 \in d} w(t', D)\, sim(t', t) \log \frac{N}{df(t)}, \text{ where}$$

w(t, D) is the initial, not expanded, weight of the term *t* in the document *D* (assigned according to TF-IDF weighting scheme in our case); *w'(t, D)* is the modified weight of the term *t* in the document *D; t'* iterates through all *(*possibly repeating) terms in the document *D ;* a is the adjustment factor (a parameter controlled in the expansion process).

4 Experiments

4.1 Similarity Error Reduction

Since in this study we were primarily concerned with improving similarity computation but not retrieval per se, we chose a widely used for text categorization Reuters collection (Lewis, 1997) over TREC or similar collections with relevance judgments. We used a modified version of Lewis' (1992) suggestion to derive our evaluation metric, which is similar to the metric derived from Kruskal-Goodman statistics used in Haveliwala et al. (2002) for a study with Yahoo web directory (www.yahoo.com). Intuitively, the metric reflects the probability of algorithm guessing the correct order (called *ground truth*), imposed by a manually created hierarchy (simplified to a partition in Reuters case). Ideally, for each document *D,* the similarity computation algorithm should indicate documents sharing one or more Reuters categories with document D to be more similar to the document *D* than the documents not sharing any categories with *D*. We formalized this intuitive requirement into a metric by the following way. Let's define a test set *Sa* to be the set of all the document triples *(D, D1, D2)* such that *D¹D1, D¹D2, D1¹D2,* and furthermore *D* shares at least one common category with *D1* but no common categories with *D2*. We defined *total error count* (*Ec*) as the number of triples in the test set *Sa* such that *sim(D, D1) < sim(D, D2)* since it should be the other way around. Our accuracy metric reported below is the total error count normalized by the size of the test set *Sa*: *similarity error = Ec / #Sa,* computed for each Reuters topics and averaged across all of them. The metric ranges from 0 (ideal case) to .5 (random ordering). It also needed an adjustment to provide the necessary continuity as justified in the following. Since the documents are represented by very sparse vectors, very often (about 5% of all triples) documents *D, D1, D2* do not have any terms in common and as a result similarity computation results in a tie: *sim(D,D1) = sim (D, D2)*. A tie can not be considered an error because in that case one can suggest a trivial improvement to the similarity algorithm by simply breaking the ties at random in any direction with an equal chance, and thus reducing errors in 50% of all ties. This is why the metric counts half of all the ties as errors, which completely removes this discontinuity.

We used all the Reuters *78* topics from the "commodity code" group since they are the most "semantic", not trying the others (Economic Indicator Codes, Currency Codes, Corporate Codes). We discarded the topics that had only 1 document and used only the documents that had at least one of the topics. This reduced our test collection to *1841* documents, still statistically powerful and computationally demanding since millions of triples had to be considered (even after some straightforward algorithmic optimizations). After indexing and stemming (Porter, 1980) the total number of unique stems used for the vector representation was *11461*.

Weighting Scheme	boolean vectors	TF only	IDF only	TF-IDF
Similarity Error	0.1750	0.1609	0.1278	0.1041

Table 2. Comparison of different weighting schemes with the original (not expanded) documents.

Table 2 lists the similarity error averaged by topics for the different weighting schemes we tried first in our experiment. Since TF-IDF weighting was by far the best in this evaluation set up, we limited our expansion experiments to TF-IDF scheme only. For similarity measure between document vectors, we used the most common negative Euclidian distance after normalizing the vectors to unit length. It can be shown, that cosine metric (dot product), the other popular metric, results in the same order and, thus same similarity error as well. Without normalization or stemming the errors were almost twice as much larger.

Although we varied the adjustment parameter *a* in our experiment, for better interpretation, we plotted our primary metric (average error reduction) as a function of *Ca*, the average Euclidian distance between the original and the modified document vectors when both vectors are normalized to unit length. *Ca* serves as a convenient parameter controlling the degree of change in the document vectors, better than *a*, because same values of *a* may result in different changes depending on the term-to-term similarity matrix *sim(t1, t2)*. In theory, *Ca* varies from *0* (no change) to $\sqrt{2}$, the case of maximum possible change (no common terms between initial and expanded representation). By varying adjustment factor *a* from 0 to 10 and higher we observed almost the entire theoretical range of *Ca*: starting from negligible change and going all the way to $\sqrt{2}$, where the added terms entirely dominated the original ones. The average number of terms in the document representation was in 60-70 range before expansion and in 200-300 range after the expansion. This of course increased computational burden. Nevertheless, even after the expansion, the vector representations still remained sparse and we were able to design and implement some straightforward algorithmic improvements taking advantage of this sparsity to keep processing time manageable. The expansion for entire Reuters collection was taking less than one minute on a workstation with Pentium III 697 MHz processor, 256 MB of RAM, with all the sparse representations of the documents and similarity matrix stored in primary memory. This renders the expansion suitable for online processing.

To evaluate the performance of each technique, we used the error reduction (%) relatively to the baseline shown in Table 1 (TF-IDF column) averaged across all the topics, which corresponds to the lowest

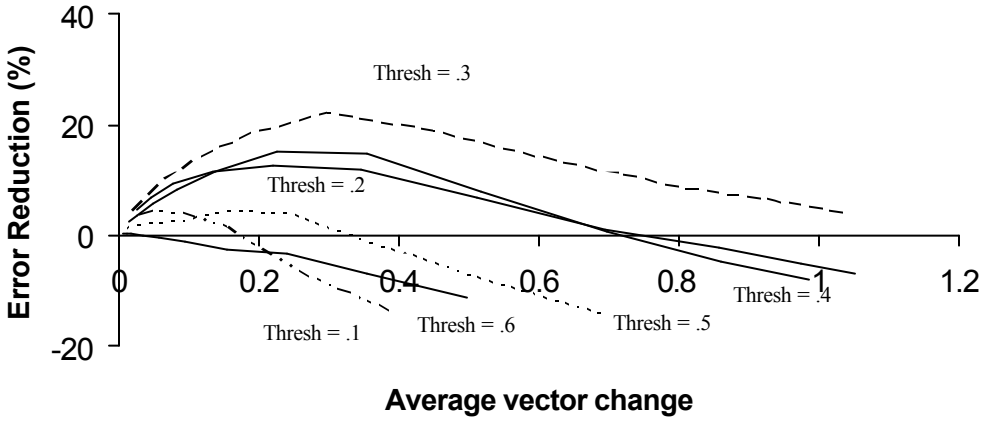

Figure 3. The average error reduction (%) as a function of average document vector change *Ca* for various threshold parameters *Tresh* without using external mining collection.

original non-expanded similarity error. Figure 1 shows the error reduction as a function of *Ca* for various values of *Thresh*. We stopped increasing *Ca* once the improvement dropped below -10% to save testing time. Several facts can be observed from the results:

1) The error reduction for *Thresh* in the mid range of *Ca [.2-.4]* is very stable, achieves *50%* , which is very large compared with the other known techniques we used for comparison as discussed below. The effect is also comparable with the difference between various weighting functions (Table 2), which we believe renders the improvement practically significant.

2) For small thresholds (*Thresh* < *.1*), the effect is not that stable, possibly since many non-reliable associations are involved in the expansion.

3) Larger thresholds (*Thresh* > *.4*) are also not very reliable since they result in a small number of associations created, and thus require large values of adjustment parameter *a* in order to produce substantial average changes in the document vectors (*Ca*), which results in too drastic change in some document vectors.

4) The error reduction curve is unimodal: it starts from 0 for small *Ca*, since document vectors almost do not change, and grows to achieve maximum for *Ca* somewhere in relatively wide *.1 - .5* range. Then, it decreases, because document vectors may be drifting too far from the original ones, falling below *0* for some large values of *Ca*.

5) For thresholds (*Thresh*) *.2* and *.3*, the effect stays positive even for large values of *Ca*, which is an interesting phenomenon because document vectors are getting almost entirely replaced by their expanded representations.

Some sensitivity of the results with respect to the parameters *Thresh, Ca* is a limitation as occurs similarly to virtually all modern IR improvement techniques. Indeed, Latent Semantic Indexing (LSI) needs to have number of semantic axis to be correctly set, otherwise the performance may degrade. Pseudo Relevance Feedback (PRF) depends on several parameters such as number of documents to use for feedback, adjustment factor, etc. All previously studied expansion techniques depend on the adjustment factor as well. The specific choice of the parameters for real life applications is typically performed manually based on trial and error or by following a machine learning approach: splitting data into training and testing sets. Based on the above results, the similarity threshold (*Thresh*) in *.2-.4* and *Ca* in *.1-.5* range seem to be a safe combination, not degrading and likely to significantly (*20-50%*) improve performance. The performance curve being unimodal with respect to both *Ca* and *Thresh* also makes it easier to tune by looking for maxima. Although we have involved only one test collection in this study, this collection (Reuters) varies greatly in the content and the size of the documents, so we hope our results will generalize to other collections.

We also verified that the effect typically diminishes when the size of the mining collection (corpus) is reduced by random sub-sampling. Those results were also similar to those obtained 4 months earlier, although only 80% of the pages in the mining corpus remained.

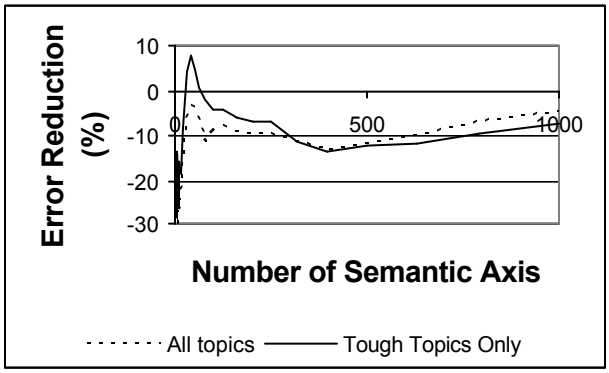

Figure 4. Comparing to LSI.

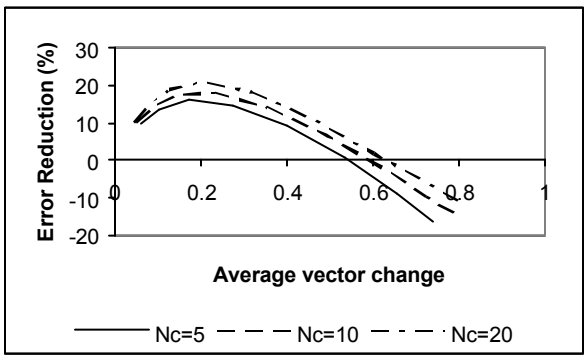

Figure 5. The error reduction as the function of the average vector change due to Pseudo Relevance Feedback for several cut-off numbers Nc.

4.2 Sensitivity Analysis

To test the importance of the context, we removed the "context hint" terms from the queries used by our agent, and created another (less context specific) corpus for mining. We obtained *175,336* unique URLs, much more than with using "context hint" terms since the overlap between different query results was much smaller. We randomly selected *25,000 URLs* of them and downloaded the referred pages. Then, to make the comparison more objective, we randomly selected *19,198* pages (same number as with using context hint) of the non-empty downloaded pages. We mined the similarity relationships from the selected documents in the same way as described above. The resulting improvement (shown in the Figure 2) was indeed much smaller (*13%* and less) than with using "context hint" terms. It also degrades much quicker for larger *Ca* and more sensitive to the choice of *Thresh*. This may explain why mixed results were reported in the literate when the similarity thesaurus was constructed in a very general setting, but not specifically for the target collection in mind. It is also interesting to note a similar behavior of error reduction as the function of *Ca* and *Thresh*: it is unimodal with maximum in approximately same range of arguments. This may also serve as indirect evidence of stability of the effect (even if smaller in that case) with respect to the parameters involved.

To verify the importance of using external corpus vs. self-mining, we mined the similarity relationships from the same collection (Reuters) that we used for the tests (target collection) using the same mining algorithms. Figure 3 shows that the effect of such "self-mining" is relatively modest (up to 20%), confirming that using the external corpus (the Web in our approach) was crucial. Again, the behavior of the error reduction (even smaller in that case) with respect to *Ca* and *Thresh* is similar to the context specific web corpus mining.

4.3 Comparison with Other Techniques

Figure 4 shows the similarity error reduction as a function of the number of semantic axis when LSI is applied. The effect with the entire collection (second column) is always negative. So, the Reuters collection in our experiment set up was found to be not a good application of LSI technique, possibly because many of the topics have already small errors even before applying LSI. To verify our implementation and the applicability of LSI to the similarity computation, we applied it only to the "tougher" 26 topics, those in the upper half if ordered by the original similarity error. As Figure 4 reveals, LSI is effective in that case for numbers of semantic axis comparable with number of topics in the target collection. Our findings are well in line with reported in prior research.

We adapted the classic Pseudo Relevance Feedback algorithm (Qiu, 1993), which has been so far applied only to document retrieval tasks, to similarity computation in a straightforward way and also tried several variations of if (not described here due to lack of space). Figure 5 shows the effect as a function of adjustment factor *a* for various cut-off parameters *Nc* (the number of top ranked documents used for feedback*)*. The effect achieves the maximum of around *21%*, consistent with the results reported in prior research. The improvement is close in magnitude to the one due to "self-mining" described above. We do not claim that our approach is better than PRF since it is not entirely meaningful to make this comparison due to the number of parameters and implementation details involved in both. Also, more important, the techniques rely on different source of data: PRF is a "self-mining" approach while CSSE builds and mines external corpus. Thus, CSSE can be used in addition to PRF.

5 Conclusions

In this paper, we proposed and empirically studied an approach to improve similarity computation between text documents by creating a context specific Web corpus and performing similarity mining within it. The results demonstrated that the similarity errors can be reduced by additional 50% after all the standard procedures such as stemming, term weighting, and vector normalization. We also established the crucial importance of the following three factors, which we believe make our technique distinct from those already explored early and explain more encouraging results that we obtained: *1) Using external corpus. 2) Taking the context of the target collection into consideration. 3) Using the appropriate mining formula*. Another important distinction and possible explanation of a more dramatic effect is our focus on similarity computation between text documents, rather than on document retrieval tasks, which have been more extensively studied in the past. Similarity computation is a more general procedure, which in turns defines the quality of virtually all other specific tasks such as document retrieval, summarization, clustering, categorization, topic detection, query by example, etc. Our future plans are to overcome some of the limitations in this study, specifically using more than a single (although standard and very diverse) collection and study other experimental setups, such as document retrieval, text categorization, or topic detection and tracking.

6 Acknowledgement

Weiguo Fan's work is supported by NSF under the grant number ITR0325579.

References

Church, K.W., Gale, W., Hanks, P., Hindle, D. (1991). Using Statistics in Lexical Analysis. In: Uri Zernik (ed.), Lexical Acquisition: Exploiting On-Line Resources to Build a Lexicon. New Jersey: Lawrence Erlbaum, 1991, pp. 115-164.

Church, K.W., Hanks, P. (1989). Word Association Norms, Mutual Information and Lexicography. *In Proceedings of the 27th Annual Conference of the Association of Computational Linguistics*, 1989, pp. 76-83.

Cooley, R., Mobasher, B. and Srivastava, J. (1997). Web Mining: Information and Pattern Discovery on the World Wide Web (with R. Cooley and J. Srivastava), in *Proceedings of the 9th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'97)*, November 1997.

Croft, W.B., and Harper, D.J. (1979). Using probabilistic models of document retrieval without relevance information. Journal of Documentation, 35, pp. 285-295.

Deerwester S., Dumais S., Furnas G., Landauer T.K., and Harshman R., Indexing by Latent Semantic Analysis. *Journal of the American Society for Information Science* 41 (1990), 391-407.

Furnas, G. W., Landauer, T. K., Gomez, L. M., & Dumais, S. T. (1987). The Vocabulary Problem in Human-System Communication. *Communications of the ACM*, 30(11), pp. 964-971.

Géry, M., Haddad, M. H. (1999). Knowledge Discovery for Automatic Query Expansion on the World Wide Web. International Workshop on the World-Wide Web and Conceptual Modeling (WWWCM'99), in conjunction with the 18th International Conference on Conceptual Modeling (ER'99), Paris, France, November 15-18, 1999, pp. 334-347.

Grefenstette, G. (1994). Explorations in Automatic Thesaurus Discovery. Kluwer Academic Publishers, Moston, MA.

Haveliwala, T.H, Gionis, A., Klein, D., Indyk, P. (2002). Evaluating Strategies for Similarity Search on the Web. WWW2002, May 7-11, 2002, Honolulu, Hawaii, USA.

Hearst, M. (1992). Automatic Acquisition of Hyponyms from Large Text Corpora, *Proceedings of the Fourteenth International Conference on Computational Linguistics*, Nantes, France, July 1992.

Kwok, K.L. (1998). Improving two-stage ad-hoc retrieval for short queries. *Twenty-First Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 250-256, New York, August 1998.

Lewis, D.D. (1992). Representation and Learning in Information Retrieval. *Doctoral Dissertation.* University of Massachusetts at Amherst.

Lewis, D.D. (1997). Reuters-21578 text categorization test collection, Distribution 1.0, Sept 26, 1997.

Minker, J., Wilson, G. A. & Zimmerman, B. H. (1972). An evaluation of query expansion by the addition of clustered terms for a document retrieval system. *Information Storage and Retrieval*, pp. 329-348.

Peat, H. J. & Willett, P. (1991). The limitations of term co-occurrence data for query expansion in document retrieval systems. *Journal of the American Society for Information Science*, 42(5), pp. 378-383.

Porter, M.F. (1980). An algorithm for suffix stripping. *Program*, 14, pp. 130--137, 1980.

Qiu, Y. (1993). Concept Based Query Expansion. Proceedings of SIGIR-93, 16th ACM International Conference on Research and Development in Information Retrieval.

Salton, G. and McGill, M.J. (1983). *Introduction to Modern Information Retrieval.* New York. McGraw-Hill.

Schutze, H. and Pedersen, J.O. (1997). A co-occurrence-based thesaurus and two applications to information retrieval. *Information Processing and Management.* 33(3), pp. 307-318.

Sugiura, A., and Etzioni, O. (2000). Query Routing for Web Search Engines: Architecture and Experiments. *9th International World Wide Web Conference*, Amsterdam, May 15-19, 2000.

van Rijsbergen, C.J.. (1977). A theoretical basis for the use of co-occurrence data in information retrieval. *Journal of Documentation*, 33(2):106--119, 1977.

Voorhees, E. M. (1994). Query expansion using lexical-semantic relations. *In Proceedings of the 17th Annual International ACM/SIGIR Conference*, pp. 61-69, Dublin, Ireland.

Xu, J. and Croft, W.B. (2000). Improving the effectiveness of information retrieval with local context analysis. ACM Transactions on Information Systems (TOIS), 18(1):79--112, 2000.

Ballesteros, L., Croft, W.B. (1998). Resolving Ambiguity for Cross-Language Retrieval. *In Proceedings of the 21th Annual International ACM/SIGIR Conference*, pp. 64-71.