Term Representation with Generalized Latent Semantic Analysis

Irina Matveeva and **Gina-Anne Levow**
Department of Computer Science, the University of Chicago
Chicago, IL 60637
Ayman Farahat and **Christiaan Royer**
Palo Alto Research Center
Palo Alto, CA 94304
{matveeva,levow}@cs.uchicago.edu {farahat,royer}@parc.com

Abstract

Document indexing and representation of term-document relations are very important issues for document clustering and retrieval. In this paper, we present Generalized Latent Semantic Analysis as a framework for computing semantically motivated term and document vectors. Our focus on term vectors is motivated by the recent success of co-occurrence based measures of semantic similarity obtained from very large corpora. Our experiments demonstrate that GLSA term vectors efficiently capture semantic relations between terms and outperform related approaches on the synonymy test.

1 Introduction

Document indexing and representation of term-document relations are crucial for document classification, clustering and retrieval (Salton & McGill 83; Ponte & Croft 98; Deerwester *et al.* 90). Since many classification and categorization algorithms require a vector space representation for the data, it is often important to have a document representation within the vector space model approach (Salton & McGill 83). In the traditional bag-of-words representation (Salton & McGill 83) of the document vectors, words represent orthogonal dimensions which makes an unrealistic assumption about the independence of terms within documents.

Modifications of the representation space, such as representing dimensions with distributional term clusters (Bekkerman *et al.* 03) and expanding the document and query vectors with synonyms and related terms as discussed in (Levow *et al.* 05), improve the performance on average. However, they also introduce some instability and thus increased variance (Levow *et al.* 05). The language modelling approach (Salton & McGill 83; Ponte & Croft 98; Berger & Lafferty 99) used in information retrieval uses bag-of-words document vectors to model document and collection based term distributions.

Since the document vectors are constructed in a very high dimensional vocabulary space, there has also been a considerable interest in low-dimensional document representations. Latent Semantic Analysis (LSA) (Deerwester *et al.* 90) is one of the best known dimensionality reduction algorithms used in information retrieval. Its most appealing features are the ability to interpret the dimensions of the resulting vector space as semantic concepts and the fact that the analysis of the semantic relatedness between terms is performed implicitly, in the course of a matrix decomposition. LSA often does not perform well on large heterogeneous collections (Ando 00). Different related dimensionality reduction techniques proved successful for document clustering and retrieval (Belkin & Niyogi 03; He *et al.* 04; Callan *et al.* 03).

In this paper, we introduce Generalized Latent Semantic Analysis (GLSA) as a framework for computing semantically motivated term and document vectors. As opposed to LSA and other dimensionality reduction algorithms which are applied to documents, we focus on computing term vectors; document vectors are computed as linear combinations of term-vectors. Thus, unlike LSA (Deerwester *et al.* 90), Iterative Residual Rescaling (Ando 00), Locality Preserving Indexing (He *et al.* 04) GLSA is not based on bag-of-words document vectors. Instead, we begin with semantically motivated pair-wise term similarities to compute a representation for terms. This shift from dual document-term representation to term representation has the following motivation.

Terms offer a much greater flexibility in exploring similarity relations than documents. The availability of large document collections such as the Web offers a great resource for statistical approaches. Recently, co-occurrence based measures of semantic similarity between terms have been shown to improve performance on such tasks as the synonymy test, taxonomy induction, and document clustering (Turney 01; Terra & Clarke 03; Chklovski & Pantel 04; Widdows 03). On the

other hand, many semi-supervised and transductive methods based on document vectors cannot yet handle such large document collections and take full advantage of this information.

In addition, content bearing words, i.e. words which convey the most semantic information, are often combined into semantic classes that correspond to particular activities or relations and contain synonyms and semantically related words. Therefore, it seems very natural to represent terms as low dimensional vectors in the space of semantic concepts.

In this paper, we use a large document collection to extract point-wise mutual information, and the singular value decomposition as a dimensionality reduction method and compute term vectors. Our experiments show that the GLSA term representation outperforms related approaches on term-based tasks such as the synonymy test.

The rest of the paper is organized as follows. Section 2 contains the outline of the GLSA algorithm, and discusses the method of dimensionality reduction as well as the term association measures used in this paper. Section 4 presents our experiments, followed by conclusion in section 5.

2 Generalized Latent Semantic Analysis

2.1 GLSA Framework

The GLSA algorithm has the following setup. We assume that we have a document collection C with vocabulary V. We also have a large Web based corpus W.

1. Construct the weighted term-document matrix D based on C
2. For the vocabulary words in V, obtain a matrix of pair-wise similarities, S, using the large corpus W
3. Obtain the matrix U^T of a low dimensional vector space representation of terms that preserves the similarities in S, $U^T \in R^{k \times |V|}$
4. Compute document vectors by taking linear combinations of term vectors $\hat{D} = U^T D$

The columns of $\hat{D}$ are documents in the k-dimensional space.

The motivation for the condition on the low dimensional representation in step 3 can be explained in the following way. Traditionally, cosine similarity between term and document vectors is used as a measure of semantic association. Therefore, we would like to obtain term vectors so that their pair-wise cosine similarities correspond to the semantic similarity between the corresponding vocabulary terms. The extent to which these latter similarities can be preserved depends on the dimensionality reduction method. Some techniques aim at preserving all pair-wise similarities, for example, the singular value decomposition used in this paper. Some graph-based approaches, on the other hand, preserve the similarities only locally, between the pairs of most related terms, e.g. Laplacian Eigenmaps Embedding (Belkin & Niyogi 03), Locality Preserving Indexing (He *et al.* 04).

The GLSA approach can combine any kind of similarity measure on the space of terms with any suitable method of dimensionality reduction. The traditional term-document matrix is used in the last step to provide the weights in the linear combination of term vectors.

In step 2, it is possible to compute the matrix S for the vocabulary of the large corpus W and use the term vectors to represent the documents in C. In addition to being computationally demanding, however, this approach would suffer from noise introduced by typos and infrequent and non-informative words. Finding methods of efficient filtering of the core vocabulary and keeping only content bearing words would be another way of addressing this issue. This is subject of future work.

2.1.1 Document Vectors

One of the advantages of the term-based GLSA document representation is that it does not have the out-of-sample problem for new documents. It does have this problem for new terms, but new terms appear at a much lower rate than documents. In addition, new rare terms will not contribute much to document classification or retrieval. Since the computation of the term vectors is done off-line, the GLSA approach would require occasional updates of the term representation.

GLSA provides a representation for documents that reflects their general semantics. Since GLSA does not transform the document vectors in the course of computation, the GLSA document representation can be easily extended to contain more specific information such as presence of proper names, dates, or numerical information.

2.2 Low-dimensional Representation

2.2.1 Singular Value Decomposition

In this section we outline some of the basic properties of the singular value decomposition (SVD) which we use as a method of dimensionality reduction. SVD is applied to the matrix S that contains pair-wise similarities between the vocaburaly terms.

First, consider the eigenvalue decomposition of S. Since S is a real symmetric matrix, it is diagonizable, i.e. it can be represented as

$$S = U\Sigma U^T$$

The columns of U are the orthogonal eigenvectors of S. Σ is a diagonal matrix containing the corresponding eigenvalues of S.

If in addition, S is positive semi-definite, it can be represented as a product of two matrices $S = \hat{U}\hat{U^T}$, and in this case $\hat{U} = U\Sigma^{1/2}$. This means that the entries of S, which in the GLSA case represent pair-wise term similarities, are inner products between the eigenvectors of S scaled with the corresponding eigenvalues.

The singular value decomposition of S is $S = U\bar{\Sigma}V^T$, where U and V are column orthogonal matrices containing the left and right singular vectors of S, respectively. $\bar{\Sigma}$ is a diagonal matrix with the singular values sorted in decreasing order.

Eckart and Young, see (Golub & Reinsch 71), have shown that given any matrix S and its singular value decomposition $S = U\Sigma V^T$, the matrix $S_k = U_k\Sigma_k V_k^T$ obtained by setting all but the first k diagonal elements in Σ to zero is

$$S_k = \text{argmin}_X ||S - X||_F^2,$$

where X is a matrix of rank k. The minimum is taken with respect to the Frobenius norm, where $||A||_F^2 = \sum_{ij} A_{ij}^2$.

The SVD of a symmetric matrix of pair-wise term similarities S is the same as its eigenvalue decomposition. Therefore, the method for computing a low-dimensional term representation that we used in this paper is to compute the eigenvalue decomposition of S and to use k eigenvectors corresponding to the largest eigenvalues as a representation for term vectors. Thus, the cosine similarities between the low dimensional GLSA term vectors preserve the semantic similarities in the matrix S for each pair of terms.

LSA is one special case within the GLSA framework. Although it begins with the document-term matrix, it can be shown that LSA uses SVD to compute the rank k approximation to a particular matrix of pair-wise term similarities. In the LSA case, these similarities are computed as the inner products between the term vectors in the space of documents, see (Bartell *et al.* 92) for details. If the GLSA matrix S is positive semi-definite, its entries represent inner products between term vectors in a feature space. Thus, GLSA with the eigenvalue decomposition can be interpreted as kernelized LSA, similar to the kernel PCA (Schölkopf *et al.* 98). Since S contains co-occurrence based similarities which have been shown to reflect semantic relations between terms, GLSA uses semantic kernels.

2.2.2 PMI as Measure of Semantic Association

We propose to obtain the matrix of semantic associations between all pairs of vocabulary terms using a number of well-established methods of computing collection-based term associations, such as point-wise mutual information, likelihood ratio, χ^2 test etc. (Manning & Schütze 99). In this paper we use point-wise mutual information (PMI) because it has been successfully applied to collocation discovery and semantic proximity tests such as the synonymy test and taxonomy induction (Manning & Schütze 99; Turney 01; Terra & Clarke 03; Chklovski & Pantel 04; Widdows 03). It was also successfully used as a measure of term similarity to compute document clusters (Pantel & Lin 02), and to extract semantic relations between verbs (Chklovski & Pantel 04).

The point-wise mutual information between random variables representing two words, w_1 and w_2, is computed as

$$PMI(w_1, w_2) = \log \frac{P(W_1 = 1, W_2 = 1)}{P(W_1 = 1)P(W_2 = 1)}.$$

The similarity matrix S with pair-wise PMI scores may not be positive semi-definite. Since such matrices work well in practice (Cox & Cox 01) one common approach is to use only the eigenvectors corresponding to the positive eigenvalues (Cox & Cox 01). This is the approach which we use in our experiments.

3 Related Approaches

As mentioned above, most related approaches compute a dual document-term representation based on the same document collection. Iterative Residual Rescaling (Ando 00) tries to put more weight on documents from underrepresented clusters of documents to improve the performance of LSA on heterogeneous collections. Random Indexing (Sahlgren & Coester 04) projects the document vectors on random low-dimensional vectors. Locality Preserving Indexing (He *et al.* 04) is a graph-based dimensionality reduction algorithm which preserves the similarities only locally. LPI differs from LSA due to the notion of locality, which is incorporated through a linear transformation of the term-document matrix. GLSA can be used with semantically motivated non-linear kernel matrices S.

Recent applications of LSA tried to compute term vectors using large collections. Document vectors for other collections are constructed as linear combinations of LSA term vectors. As mentioned above, LSA uses only one particular measure of term similarity. The Word Space Model for word sense disambiguation developed by Schütze (Schütze 98) is another special case of GLSA which computes term vectors directly. Instead of using document co-occurrence statistics, it uses term co-occurrence in the contexts of the most frequent informative terms, then SVD is applied. One particular kind of co-occurrence based similarities, namely normalized counts, are used (Schütze 98; Widdows 03). Latent Relational Analysis (Turney 04) looks at pair-wise relations between selected terms and not at term vectors for the whole vocabulary and uses co-occurrence counts within context patterns. SVD is applied to the matrix of similarities between the context patterns as a method of smoothing the similarity information.

The probabilistic LSA (Hofmann 99) and Latent Dirichlet Allocation (Blei *et al.* 02) use the latent semantic concepts as bottleneck variables in computing the term distributions for documents. The probabilities are estimated using the EM algorithm which can suffer from local minima and has a large space requirement. This limits the use of these approaches for large document collection.

4 Experiments

The goal of the experimental evaluation of the GLSA term vectors was to demonstrate that the GLSA vector space representation for terms captures their semantic relations. We used the synonymy and term pairs tests for the evaluation. Our results demonstrate that similarities between GLSA term vectors achieve better results than the latest approaches based on PMI scores (Terra & Clarke 03).

To collect the co-occurrence for the matrix of pair-wise term similarities S, in all experiments presented here we used the English Gigaword collection (LDC), containing New York Times articles. We only used the documents that had the label "story". Thus, we used a collection comprised of 1,119,364 documents with 771,451 terms. We used the Lemur toolkit[1] to tokenize and index all document collections used in our experiments; we used stemming and a list of stop words.

The similarities matrix S was constructed using the PMI scores. In our preliminary experiments we used some other co-occurrence based measures of similarities, such as likelihood ratio and χ^2 test but obtained results which were below those for PMI. Therefore, we do not report them here. We used the PMI matrix S in combination with SVD (denoted as $GLSA$) to compute GLSA term vectors. Unless stated otherwise, for the GLSA method we report the best performance over different numbers of embedding dimensions. We used the PLAPACK package[2] to perform the SVD (Bientinesi *et al.* 03).

4.1 Synonymy Test

The synonymy test represents a list of words and for each of them, there are 4 candidate words. The task is to determine which of these candidate words is a synonym to the word in question. This test was first used to demonstrate the effectiveness of LSA term vectors (Landauer & Dumais 97). More recently, the PMI-IR approach developed by Turney (Turney 01) was shown to outperform LSA on this task (Turney 01) and (Terra & Clarke 03).

We evaluated the GLSA term vectors on the synonymy test and compared the results to the latest results with the PMI-IR approach (Terra & Clarke 03). Terra et al. (Terra & Clarke 03) com-

[1] http://www.lemurproject.org/

[2] http://www.cs.utexas.edu/users/plapack/

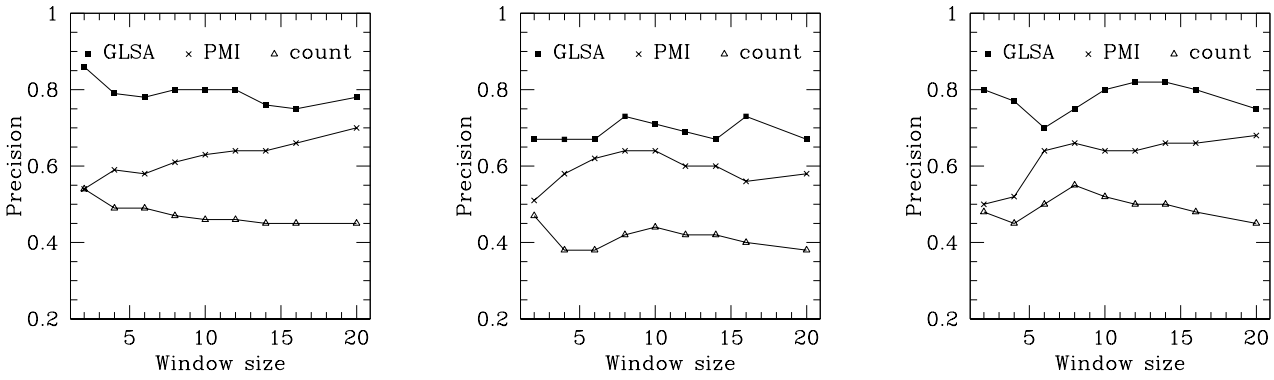

Figure 1: Precision with GLSA, PMI and count over different window sizes, for the TOEFL(left), TS1(middle) and TS2(right) tests.

pared the performance of different co-occurrence based measures of term similarity on the synonymy test and came to the conclusion that PMI yielded the best results.

Following (Terra & Clarke 03), we used the TOEFL, TS1 and TS2 synonymy tests. The TOEFL test contains 80 synonymy questions. We also used the preparation tests called TS1 and TS2. Since GLSA in its present formulation cannot handle multi-word expressions, we had to modify the TS1 and TS2 tests slightly. We removed all test questions that contained multi-word expressions. From 50 TS1 questions we used 46 and from 60 TS2 questions we used 49. Thus, we would like to stress that the comparison of our results on TS1 and TS2 to the results reported in (Terra & Clarke 03) is only suggestive. We used the TS1 and TS2 test without context. The only difference in the experimental setting for the TOEFL test between our experiments and the experiments in (Terra & Clarke 03) is in the document collections that were used to obtain the co-occurrence information.

4.1.1 GLSA Setting

To have a richer vocabulary space, we added the 2000 most frequent words from the English Gigaword collection to the vocabularies of the TOEFL, TS1 and TS2 tests. We computed GLSA term vectors for the extended vocabularies of the TOEFL, TS1 and TS2 tests and selected the term t^* whose term vector had the highest cosine similarity to the question term vector $\vec{t_q}$ as the synonym. We computed precision scores as the ratio of correctly guessed synonyms.

The co-occurrence counts can be obtained using either term co-occurrence within the same document or within a sliding window of certain fixed size. In our experiments we used the window-based approach which was shown to give better results (Schütze 98; Terra & Clarke 03). Since the performance of co-occurrence based measures is sensitive to the window size, we report the results for different window sizes.

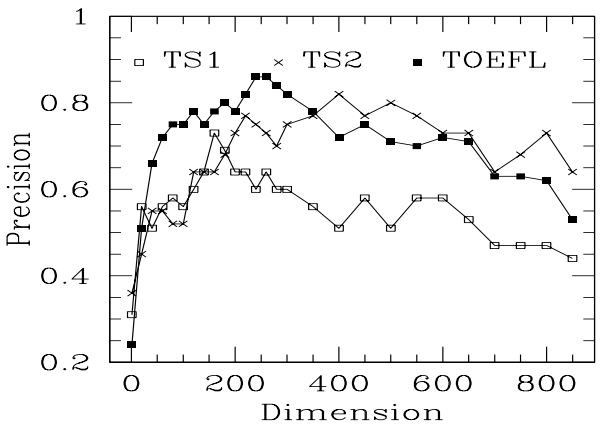

Figure 2: Precision at different numbers of GLSA dimensions with the best window size.

4.1.2 Results on the Synonymy Test

Figure 1 shows the precision using different window sizes. The baselines are to choose the candidate with the highest co-occurrence count or PMI score. For all three data sets, GLSA significantly outperforms PMI scores computed on the same collection. The results that we obtained using just the PMI score are below those reported in Terra and Clarke (Terra & Clarke 03). One explanation for this discrepancy is the size and the composition of the document collections used for the co-occurrence statistics. The English Gigaword collection that we used is smaller and, more im-

portantly, less heterogeneous than the web based collection in (Terra & Clarke 03). Nonetheless, on the TOEFL data set GLSA achieves the best precision of 0.86, which is much better than our PMI baseline as well as the highest precision of 0.81 reported in (Terra & Clarke 03). GLSA achieves the same maximum precision as in (Terra & Clarke 03) for TS1 (0.73)and a much higher precision on TS2 (0.82 vs. 0.75 in (Terra & Clarke 03)).

Figure 2 shows the precision for the GLSA terms only, using different number of dimensions. The number of dimensions is important because it is one of the parameter in the GLSA setting. LSA-based approaches usually perform best with 300-400 resulting dimensions. The variation of precision at different numbers of embedding dimensions within the 100-600 range is somewhat high for TS1 but much smoother for the TOEFL and TS2 tests.

4.2 Term Pairs Test

Some of the terms on the synonymy test are infrequent (eg. "wig") and some are usually not considered informative (eg. "unlikely"). We used the following test to evaluate how the cosine similarity between GLSA vectors captures similarity between terms which are considered important for such tasks as document classification.

We computed GLSA term vectors for the vocabulary of the 20 news groups document collection. Using the Rainbow software[3] we obtained the top N words with the highest mutual information with the class label. We also obtained the probabilities that each of these words has with respect to each of the news groups. We assigned the group in which the word has the highest probability as the word's label. Some of the top words and their labels can be seen in Table 3. Although the way we assigned labels may not strictly correspond to the semantic relations between words, this table shows that for this particular collection and for informative words (e.g., "bike","team") they do make sense.

We computed pair-wise similarities between the top N words using the cosine between the GLSA vectors representing these words and also used just the PMI scores. Then we looked at the pairs of terms with the highest similarities. Since for this test we selected content bearing words, the intuition is that most similar words should be semantically related and are likely to appear in documents belonging to the same news group. Therefore, they should have the same label. Each word can also be considered a query, and in this test we are trying to retrieve other words that are semantically most related to the it.

This task is better suited to demonstrate the advantage of GLSA over PMI-IR. In the synonymy task the comparisons are made between the PMI scores of a few carefully selected terms that are synonymy candidates for the same word. While PMI-IR performs quite well on the synonymy task, it is in general difficult to compare PMI scores across different pairs of words. Apart from this normalization issue, PMI scores for rare words tend to be very high, see (Manning & Schütze 99). Our experiments illustrate that GLSA significantly outperforms the PMI scores on this test.

We used $N = \{100, 1000\}$ top words by the MI with the class label. The top 100 are highly discriminative with respect to the news group label whereas the top 1000 words contain many frequent words. Our results show that GLSA is much less sensitive to this than PMI.

First we sort all pairs of words by similarity and compute precision at the k most similar pairs as the ratio of word pairs that have the same label. Table 1 shows that GLSA significantly outperforms the PMI score. PMI has very poor performance, since here the comparison is done across different pairs of words.

The second set of scores was computed for each word as precision at the top k nearest terms, similar to precision at the first k retrieved documents used in IR. We report the average precision values for different values of k in Table 2. GLSA achieves higher precision than PMI. GLSA performance has a smooth shape peaking at around 200-300 dimension which is in line with results for other SVD-based approaches (Deerwester *et al.* 90; He *et al.* 04). The dependency on the number of dimensions was the same for the top 1000 words.

In Table 3 we show the individual results for some of the words. GLSA representation achieves very good results for terms that are not very frequent in general document collections but are very good indicators of particular news groups, such as "god" or "bike". For much more frequent words, and words which have multiple senses,

[3] http://www-2.cs.cmu.edu/ mccallum/bow/rainbow/

	top 100		top 1000	
k	Pmi	Glsa	Pmi	Glsa
1	0.0	1.0	0.0	1.0
5	0.0	1.0	0.0	1.0
10	0.0	1.0	0.0	0.8
50	0.32	0.88	0.12	0.8
100	0.24	0.76	0.1	0.8

Table 1: Precision for the term pairs test at the top k most similar pairs.

	top 100		top 1000	
k	Pmi	Glsa	Pmi	Glsa
1	0.27	0.67	0.08	0.43
5	0.40	0.48	0.8	0.40
10	0.35	0.37	0.1	0.37
50	0.14	0.13	0.16	0.20
100	0.08	0.08	0.16	0.18

Table 2: Average precision for the term pairs test at the top k nearest words.

word	nn=1	nn=2	nn=3	Prec
god (18)	jesus (18)	bible (18)	heaven (18)	1
bike (15)	motorcycle (15)	rider (15)	biker (15)	1
team (17)	coach (17)	league (20)	game (17)	0.6
car (7)	driver (1)	auto (7)	ford (7)	0.6
windows (1)	microsoft (1)	os (3)	nt (1)	0.4
dod (15)	agency (10)	military (13)	nsa (10)	0
article (15)	publish (13)	fax (4)	contact (5)	0

Table 3: Precision at the 5 nearest terms for some of the top 100 words by mutual information with the class label. The table also shows the first 3 nearest neighbors. The word's label is given in the brackets. (1=os.windows; 3=hardware; 4=graphics; 5=forsale; 7=autos; 10=crypt; 13=middle-east;15=motorcycles; 17=hokey; 18=religion-christian; 20=baseball.)

such as "windows" or "article", the precision is lower. The pair "car", "driver" is semantically related for one sense of the word "driver", but the word "driver" is assigned to the group "windows-os" with a different sense.

5 Conclusion and Future Work

Our experiments have shown that the cosine similarity between the GLSA term vectors corresponds well to the semantic similarity between pairs of terms. Interesting questions for future work are connected to the computational issues. As other methods based on a matrix decomposition, GLSA is limited in the size of vocabulary that it can handle efficiently. Since terms can be divided into content-bearing and function words, GLSA computations only have to include content-bearing words. Since the GLSA document vectors are constructed as linear combinations of term vectors, the inner products between the term vectors are implicitly used when the similarity between the document vectors is computed. Another interesting extension is therefore to incorporate the inner products between GLSA term vectors into the language modelling framework and evaluate the impact of the GLSA representation on the information retrieval task.

We have presented the GLSA framework for computing semantically motivated term and document vectors. This framework allows us to take advantage of the availability of large document collection and recent research of corpus-based term similarity measures and combine them with dimensionality reduction algorithms. Using the combination of point-wise mutual information and singular value decomposition we have obtained term vectors that outperform the state-of-the-art approaches on the synonymy test and show a clear advantage over the PMI-IR approach on the term pairs test.

Acknowledgements We are very grateful to Paolo Bientinesi for his extensive help with adopting the PLAPACK package to our problem. The TOEFL questions were kindly provided by Thomas K. Landauer, Department of Psychology, University of Colorado. This research has been funded in part by contract #MDA904-03-C-0404 to Stuart K. Card and Peter Pirolli from the Advanced Research and Development Activity, Novel Intelligence from Massive Data program.

References

(Ando 00) Rie Kubota Ando. Latent semantic space: iterative scaling improves precision of inter-document similarity measurement. In *Proc. of the 23rd ACM SIGIR*, pages 216–223, 2000.

(Bartell *et al.* 92) Brian T. Bartell, Garrison W. Cottrell, and Richard K. Belew. Latent semantic indexing is an optimal special case of multidimensional scaling. In *Proc. of the 15th ACM SIGIR*, pages 161–167. ACM Press, 1992.

(Bekkerman *et al.* 03) Ron Bekkerman, Ran El-Yaniv, and Naftali Tishby. Distributional word clusters vs. words for text categorization, 2003.

(Belkin & Niyogi 03) Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15(6):1373–1396, 2003.

(Berger & Lafferty 99) Adam Berger and John Lafferty. Information retrieval as statistical translation. In *Proc. of the 22rd ACM SIGIR*, 1999.

(Bientinesi *et al.* 03) Paolo Bientinesi, Inderjit S. Dhilon, and Robert A. van de Geijn. A parallel eigensolver for dense symmetric matrices based on multiple relatively robust representations. *UT CS Technical Report TR-03-26*, 2003.

(Blei *et al.* 02) David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. In *Proc. of 14th NIPS*, New York, 2002. ACM.

(Callan *et al.* 03) Jamie Callan, Gordon Cormack, Charles Clarke, David Hawking, and Alan Smeaton. Document clustering based on non-negative matrix factorization. In *Proc. of the 26rd ACM SIGIR*, New York, 2003. ACM.

(Chklovski & Pantel 04) Timothy Chklovski and Patrick Pantel. Verbocean: Mining the web for fine-grained semantic verb relations. In *Proc. of EMNLP*, 2004.

(Cox & Cox 01) Trevor F. Cox and Micheal A. Cox. *Multidimensional Scaling*. CRC/Chapman and Hall, 2001.

(Deerwester *et al.* 90) Scott C. Deerwester, Susan T. Dumais, Thomas K. Landauer, George W. Furnas, and Richard A. Harshman. Indexing by latent semantic analysis. *Journal of the American Society of Information Science*, 41(6):391–407, 1990.

(Golub & Reinsch 71) G. Golub and C. Reinsch. *Handbook for Matrix Computation II, Linear Algebra*. Springer-Verlag, New York, 1971.

(He *et al.* 04) Xiaofei He, Deng Cai, Haifeng Liu, and Wei-Ying Ma. Locality preserving indexing for document representation. In *Proc. of the 27rd ACM SIGIR*, pages 96–103. ACM Press, 2004.

(Hofmann 99) Thomas Hofmann. Probabilistic latent semantic analysis. In *Uncertainty in Artificial Intelligence*, 1999.

(Landauer & Dumais 97) Thomas K. Landauer and Susan T. Dumais. A solution to platos problem: The latent semantic analysis theory of the acquisition, induction, and representation of knowledge. *Psychological Review*, 1997.

(Levow *et al.* 05) Gina-Anne Levow, Douglas W. Oard, and Philip Resnik. Dictionary-based techniques for cross-language information retrieval. *Information Processing and Management: Special Issue on Cross-language Information Retrieval*, 2005.

(Manning & Schütze 99) Chris Manning and Hinrich Schütze. *Foundations of Statistical Natural Language Processing*. MIT Press. Cambridge, MA, 1999.

(Pantel & Lin 02) Patrick Pantel and Dekang Lin. Document clustering with committees. In *Proc. of the 25th ACM SIGIR*, pages 199–206. ACM Press, 2002.

(Ponte & Croft 98) Jay M. Ponte and W. Bruce Croft. A language modeling approach to information retrieval. In *Proc. of the 21st ACM SIGIR*, pages 275–281, New York, NY, USA, 1998. ACM Press.

(Sahlgren & Coester 04) Magnus Sahlgren and Rickard Coester. Using bag-of-concepts to improve the performance of support vector machines in text categorization. In *Proc. of the 20th COLING*, pages 487–493, 2004.

(Salton & McGill 83) Gerard Salton and Michael J. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, 1983.

(Schölkopf *et al.* 98) Bernhard Schölkopf, Alex J. Smola, and Klaus-Robert Muller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10, 1998.

(Schütze 98) Hinrich Schütze. Automatic word sense discrimination. *Computational Linguistics*, 24(21):97–124, 1998.

(Terra & Clarke 03) Egidio L. Terra and Charles L. A. Clarke. Frequency estimates for statistical word similarity measures. In *Proc.of HLT-NAACL*, 2003.

(Turney 01) Peter D. Turney. Mining the web for synonyms: PMI–IR versus LSA on TOEFL. *Lecture Notes in Computer Science*, 2167:491–502, 2001.

(Turney 04) Peter D. Turney. Human-level performance on word analogy questions by latent relational analysis. Technical report, Technical Report ERB-1118, NRC-47422, 2004.

(Widdows 03) Dominic Widdows. Unsupervised methods for developing taxonomies by combining syntactic and statistical information. In *Proc. of HLT-NAACL*, 2003.