

# Document representation with Generalized Latent Semantic Analysis

Irina Matveeva

Department of Computer Science

University of Chicago

Chicago, IL 60637

matveeva@cs.uchicago.edu

Ayman Farahat

Xerox Palo Alto Research Center

3333 Coyote Hill Road

Palo Alto, CA 94304

farahat@parc.com

Christiaan Royer

Xerox Palo Alto Research Center

3333 Coyote Hill Road

Palo Alto, CA 94304

royer@parc.com

## **Abstract**

Methods for dimensionality reduction, notably LSA, have been successfully applied to the information retrieval task and document classification. Recently, corpus-based association measures such as point-wise mutual information have been found to outperform LSA on a variety of tasks. We have developed an algorithmic framework that computes a low-dimensional vector space representation of documents combining different measures of association with different dimensionality reduction techniques. Experimental results show a competitive performance on the synonymy and text classification tasks.

# 1 Introduction

There is a wide range of applications that require comparison of text data, such as information retrieval, information filtering and text classification. Here, we introduce Generalized Latent Semantic Analysis, a framework which belongs to vector space model (VSM) methods. Finding a meaningful vector space representation of text data is a difficult problem. Different approaches have been developed to overcome the term independence assumption of the traditional bag-of-words (BOW) representation [17] by such modifications of the representation space as using distributional term clusters to represent dimensions [19, 3, 2] and expanding the document vectors with synonyms and related terms, as discussed in [5, 11].

Multidimensional scaling (MDS) comprises a family of methods to compute a vector space representation for data for which only pair-wise similarities or dissimilarities are available. Latent Semantic Analysis (LSA) [8] is one of the best known MDS algorithms used in IR. One of the advantages of the MDS framework is 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 uses a particular term similarity measure for MDS. Our approach was motivated by the considerable interest in other co-occurrence based measures of semantic relations between terms [22, 21, 6]. Corpus-based association measures such as point-wise mutual information have been found to outperform LSA on a variety of tasks [22, 13].

In this paper we investigate a more general use of MDS for a low-dimensional vector space representation of vocabulary terms. Generalized Latent Semantic Analysis (GLSA) is a framework to combine different measures of semantic association between terms and different methods of dimensionality reduction.

Our approach takes advantage of recent developments in many areas: availability of large document collections such as the Web and successful use of point-wise mutual information as a term similarity measure alternative to LSA, as well as development of new graph-based dimensionality reduction algorithms. In this paper, we show that GLSA algorithms, in particular a combination of point-wise mutual information with metric MDS and Laplacian Eigenmaps Embedding, achieve competitive performance on the synonymy and text classification tasks.

## 2 Dimensionality Reduction for Document Representation

There has been a considerable amount of research on the use of dimensionality reduction techniques for document representation. We briefly review some of the related approaches to motivate our method. LSA makes the assumption that there is a latent semantic structure in a given document collection which is often obscured due to synonymy and polysemy [8]. LSA reveals the semantic structure and indexes documents not by individual terms, but by latent semantic concepts represented by them. There have been many attempts to improve LSA performance on heterogeneous collections [1, 7].

The bag-of-concepts (BOC) vector space representation of documents was recently investigated by Sahlgren et al. [16]. This representation is computed as Random Indexing (RI) by projecting the BOW document vectors on random low-dimensional vectors. Sahlgren et al. introduce a new method of iterative computation of the document vectors which overcomes the difficulties associated with the high computational cost of the LSA decomposition.

LSA and RI were developed to compute a vector space representation of documents. They can also be seen as methods of obtaining low-dimensional term vectors, but their duality often ties the computation of term vectors to a particular document collection. Recent applications of LSA, however, also concentrated on computing term vectors using large collections. Document vectors for other collections are constructed as linear combinations of LSA term vectors. Still, LSA uses only one particular measure of term similarity.

Terms offer a much greater flexibility in exploring similarity relations than documents. Semantic similarities between terms can be analyzed using any large document collection such as the Web. The Word Space Model (WSM) for word sense disambiguation developed by Schuetze [18] is another special case of MDS which computes the term vectors directly. Instead of using document co-occurrence statistics, it uses term co-occurrence in the contexts of the most frequent informative terms. Widdows used WSM vectors to represent documents for IR and taxonomy induction [23, 24]. The similarities between any given term and the most informative terms can be computed in a number of ways, but usually normalized counts are used [18, 23]. LSA and WSM differ in the way in which they obtain pair-wise term similarities. After that, the eigenvalue decomposition of the similarities

matrix is used to obtain the term vectors.

One of the important issues that we explore here is which methods of obtaining corpus-based measures of term association can be combined with MDS.

The similarity measure is not the only parameter in this setting, one can also make use of various methods for dimensionality reduction. Laplacian Eigenmaps Embedding [4], Isomap [20], and Locality Preserving Projections (LPP) [10] are examples of graph-based modifications of the MDS approach to dimensionality reduction.

We present Generalized Latent Semantic Analysis (GLSA) as a framework for computing vector space representation of term vectors. This framework generalizes the use of dimensionality reduction in two ways. First, we consider different measures of semantic association between terms and use co-occurrence counts computed using large resources such as Web-based collections and the Web. Secondly, we combine them with different dimensionality reduction techniques such as the graph-based Laplacian Eigenmaps Embedding algorithm.

All algorithms within the GLSA framework deal with synonymy and polysemy in the same way as LSA, by providing a similar notion of extracting latent semantic concepts. For the term association measures we consider here, GLSA will compute a good embedding under the same assumptions as LSA [14].

The rest of the paper is organized as follows. Section 3 contains the outline of the GLSA algorithm, and discusses the methods of dimensionality reduction as well as the term association measures used in this paper. Section 4 presents our IR experiments, followed by discussion in section 5.

## 3 Generalized Latent Semantic Analysis

### 3.1 GLSA Algorithm

The GLSA algorithm has the following steps. We assume that we have a document collection  $C$  with  $N$  documents, with vocabulary size  $V$ . We also have a large Web based corpus  $W$ .

1. Construct the weighted term document matrix  $D_{V \times N}$  using  $C$
2. Obtain a matrix of pair-wise similarities  $S_{V \times V}$  using  $W$

3. Obtain a low dimensional vector space representation of terms that preserves the similarities in  $S$ ,  $U_{k \times V}^T$
4. Compute document vectors by taking linear combinations of term vectors

$$\hat{D} = U_k^T D$$

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

It is evident from the above description that this approach can be combined with any kind of similarity measure on the space of terms as well as any 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. We use entropy-based weights which have proven very effective in a number of applications, e.g. [9].

## 3.2 Measure of Semantic Association

To be able to apply the metric MDS as well as other dimensionality reduction techniques, we first have to obtain a matrix of semantic association between all pairs of vocabulary terms. There is a number of well-established methods to compute collection-based term associations, such as point-wise mutual information, likelihood ratio,  $\chi^2$  test etc. [12]. These measures of pair-wise term associations have been successfully applied to collocation discovery and semantic proximity tests [12, 22, 21, 6, 24]. In the synonymy test [22, 21] these measures are applied to a small number of carefully selected term pairs. This is quite different from the GLSA situation when all pair-wise comparisons are needed. Therefore, we were interested in exploring the possibility of using these similarity measures to compute the GLSA similarities matrix.

We briefly describe the association measures that we used in our experiments. We write the number of co-occurrences of terms  $x$  and  $y$  as  $C(x, y)$ , the number of term occurrences as  $C(x)$  and  $C(y)$ , and the total number of documents is  $N$ . The measures that we used serve as a comparison test between the assumption that the terms occur independently and the assumption that they are dependent.

### 3.2.1 $\chi^2$ test

The values assigned by this test are obtained from the 2-dimensional contingency table associated with each term pair. Following Schuetze, [18],

we compute the following value:

$$\chi^2 = \frac{N(N_{++}N_{--} - N_{+-}N_{-+})^2}{(N_{++} + N_{+-})(N_{-+}N_{--})(N_{++}N_{-+})(N_{+-}N_{--})},$$

where  $N_{++} = C(x, y)$ ,  $N_{+-} = C(x) - C(x, y)$ ,  $N_{-+} = C(y) - C(x, y)$  and  $N_{--} = N - N_{++} - N_{+-} - N_{-+}$ .

### 3.2.2 Log Likelihood ratio

In this case the hypothesis of independence is tested under the assumption of the binomial term distribution.  $H1$  is that  $P(x|y) = P(x|\neg y)$  and  $H2$  is that  $P(x|y) \neq P(x|\neg y)$ . These hypotheses determine the parameters of the binomial distribution and so the likelihood  $L(H1)$  and  $L(H2)$  of the data under  $H1$  and  $H2$  can be computed (see [12] for details).

$$\log \lambda = \log \frac{L(H1)}{L(H2)}.$$

Following the discussion in [12], we used the  $\chi^2$ -distributed value of  $-2 \log(\lambda)$ .

### 3.2.3 Odds ratio

The odds ratio computes the ratio of the number of times that the words occur together to the number of times the words occur individually.

$$\text{odds ratio} = \frac{C(x, y)}{C(x)C(y)}$$

### 3.2.4 Point-wise mutual information

If we take a pair of vocabulary terms,  $x$  and  $y$  and map them into binary random variables,  $X$  and  $Y$ , we can compute their point-wise mutual information as the amount of information that an occurrence of  $x$  contains about the occurrence of  $y$ . The PMI similarity is computed as

$$PMI(x, y) = \log \frac{P(X = 1, Y = 1)}{P(X = 1)P(Y = 1)}.$$

### 3.3 PMI-based score

Turney [22] used statistical information from the Web to compute point-wise mutual information (PMI) between vocabulary terms and obtained better results on the synonyms test than LSA. In this paper, we are particularly interested in PMI as a term similarity measure because it has a number of successful applications. Lin and Pantel [13] used PMI as a measure of term similarity to compute document clusters, Chklovski and Pantel [6] use web-based PMI to extract semantic relations between verbs. It also has been shown to perform better than other statistical measures of association. Terra and Clarke [21] studied a number of different measures of word association computed using a terabyte-size corpus of Web data applied to the synonyms task and showed that PMI achieves the best results.

These researchers compared the performance of the synonymy extraction task using the PMI score to that of using the cosine between the LSA term vectors. As mentioned above, the LSA similarity measure is the inner product between term vectors in the space of documents. PMI can also be seen as the log of appropriately normalized term vectors in the same space. However, LSA also performs a dimensionality reduction which denoises the matrix of pair-wise similarities. Nevertheless, a PMI-based score leads to better performance [22, 21]. Thus, we explore the possibility of a similar transformation of PMI-based similarities in the GLSA framework.

## 4 Experiments

### 4.1 Synonymy Test

To have a quantitative evaluation of the GLSA term vectors, we first used them on the synonymy test. Other researchers used this test for the evaluation of the LSA and PMI-IR approaches, and we use their results for comparison.

We used the TS1 and TS2 synonyms test. In the current version of our system we do not handle multi-word expressions. Therefore, we removed the questions that contained such expressions from our test set. The resulting TS1 test set contained 48 synonymy questions, the TS2 test had 49 questions.

### 4.1.1 GLSA Collection

We used the English Gigaword collections (LDC), containing New York Times articles from July 1994 to June 2002. We only used the documents that had the label “story”; 1,119,364 documents with 771,451 terms.

### 4.1.2 Experimental Setting

We computed the pair-wise co-occurrence counts for all terms in the synonyms test list using different window sizes. Then we computed the matrix of PMI-based similarities and then the GLSA term vectors. We computed the inner product between the term vectors in the same synonymy question (between the term in question and the candidates) and assigned the term with the highest similarity to be the answer. In our preliminary experiments we first computed the GLSA vectors using only the words from the tests. However, the vocabulary size was very small (around 250 words) and the results were not much different from the PMI score based results. After that we extended the space of the vocabulary for which the GLSA vectors were computed to include 2000 most frequent terms from the English Gigaword collection. This approach is similar to the Schuetze’s WordSpace approach [18] but we used the primary similarity information between pairs of terms for the following decomposition. The idea was that larger vocabulary space gives a better similarity information. Below we report the results based on the enlarged vocabulary spaces. The vocabulary size for the TS1 test is 2128, for the TS2 it is 2120. We also tried to extend the vocabulary using part of speech information and added only 2000 most frequent nouns as well as add more terms, but it did not affect the results significantly and we do not report them here.

### 4.1.3 Results

We computed the precision of the synonymy extraction test using TS1 and TS2 collection using different parameters for computing the GLSA vectors. As mentioned before, Terra and Clarke [21] performed similar experiments using only co-occurrence based association measures such as PMI,  $\chi^2$  and log likelihood ratio. They achieved best results when the co-occurrence counts were computed using a sliding window of a certain size and PMI was used as a measure of term association. To be able to compare to their results, we computed the precision on TS1

| Window Size | PMI(T&C) | PMI  | GLSA |
|-------------|----------|------|------|
| 2           | 0.73     | 0.51 | 0.68 |
| 4           | 0.68     | 0.59 | 0.68 |
| 8           | 0.66     | 0.63 | 0.73 |
| 16          | 0.66     | 0.59 | 0.73 |
| 20          | NA       | 0.59 | 0.69 |
| 32          | 0.70     | NA   | NA   |

Table 1: Precision for the TS1 synonyms test. The maximum precision obtained by Terra and Clarke is 73% (with window size 2).

and TS2 tests with different window sizes and used PMI-based GLSA to compute the term vectors. Tables 1, 2 show the precision on each test for three different settings. We computed the precision using only the PMI score, similar to Terra and Clarke [21], the results are shown in the column labeled PMI. Then we computed the precision using the  $GLSA_{PMI}$  term vectors, these results are shown in the column labeled GLSA. In the columns labeled PMI(T&C) we show the best results that were reported in Terra and Clarke [21] for comparison. We would like to stress here that the experimental setting in this paper and in Terra and Clarke paper are somewhat different. We used a different corpus, of a different size and composition. We also removed some of the questions from the test. So that the comparison of results is only suggestive.

From tables 1, 2 it can be seen that GLSA achieves the same precision as Terra and Clarke [21] for TS1 and a much higher precision on TS2. However, the results that we obtained using just the PMI score are below those reported in Terra and Clarke [21]. GLSA significantly outperforms the PMI scores computed on the same collection.

Figures 1, 2 summarize the precision using different window sizes. For GLSA here we report the best precision over different numbers of embedding dimensions. Figures 3, 4 show the precision for the GLSA terms only, using different number of dimensions. It can be seen that GLSA outperforms both baselines at all window sizes that we tried. The variation of precision at different numbers of embedding dimensions is somewhat high for TS1 but much smoother for the TS2 test.

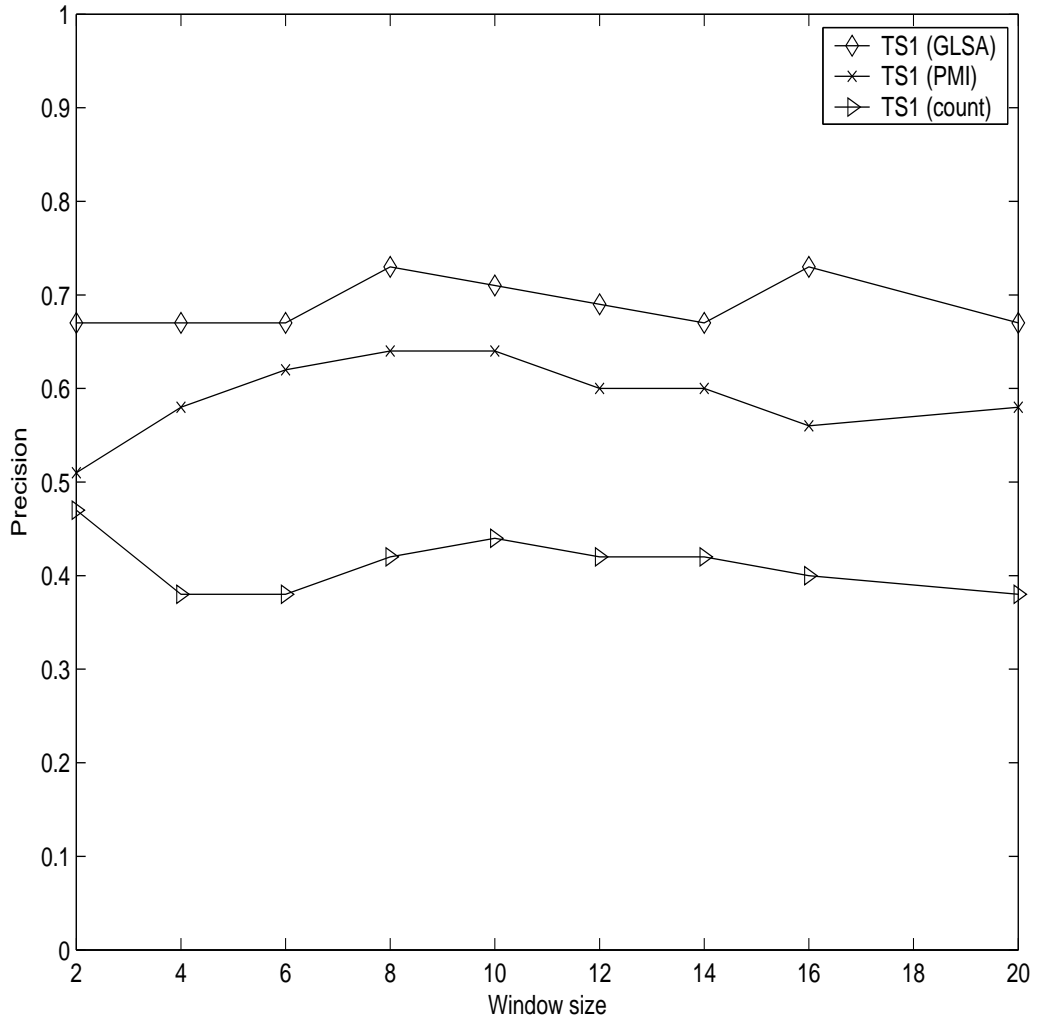


Figure 1: TS1 test. Precision is computed using the co-occurrence count, the PMI score and the similarity between the GLSA vectors to find the synonym.

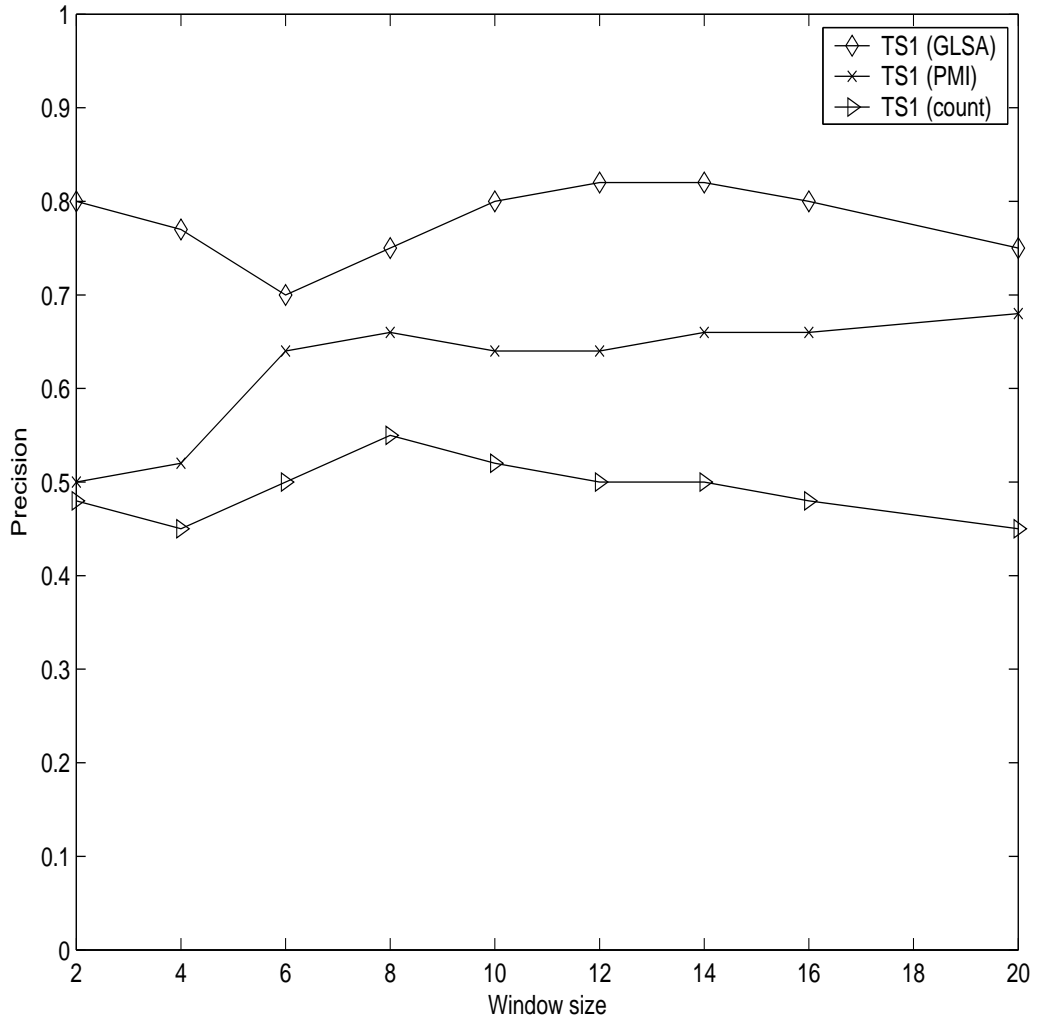


Figure 2: TS2 test. Precision is computed using the co-occurrence count, the PMI score and the similarity between the GLSA vectors to find the synonym.

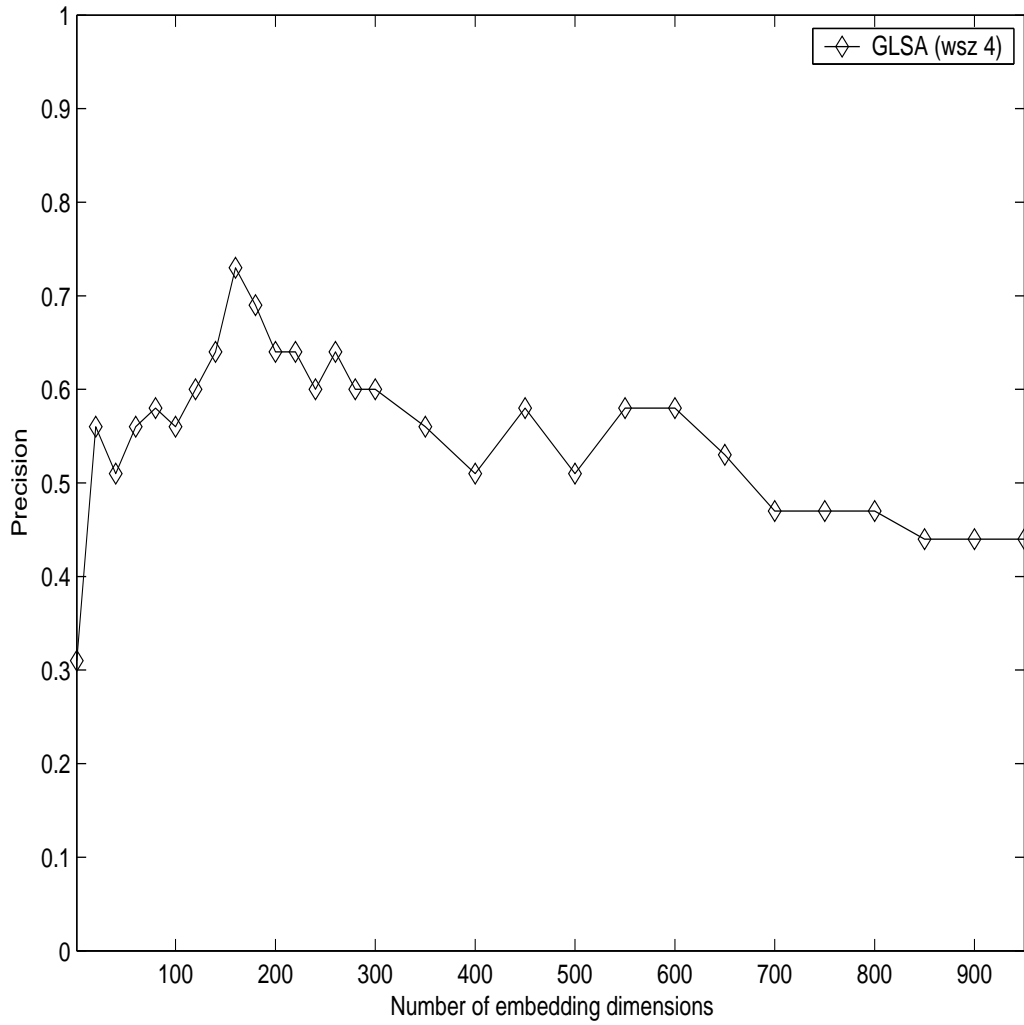


Figure 3: TS1 test. Measure = GLSA(pmi). Window size is 4. Precision at different numbers of GLSA dimensions.

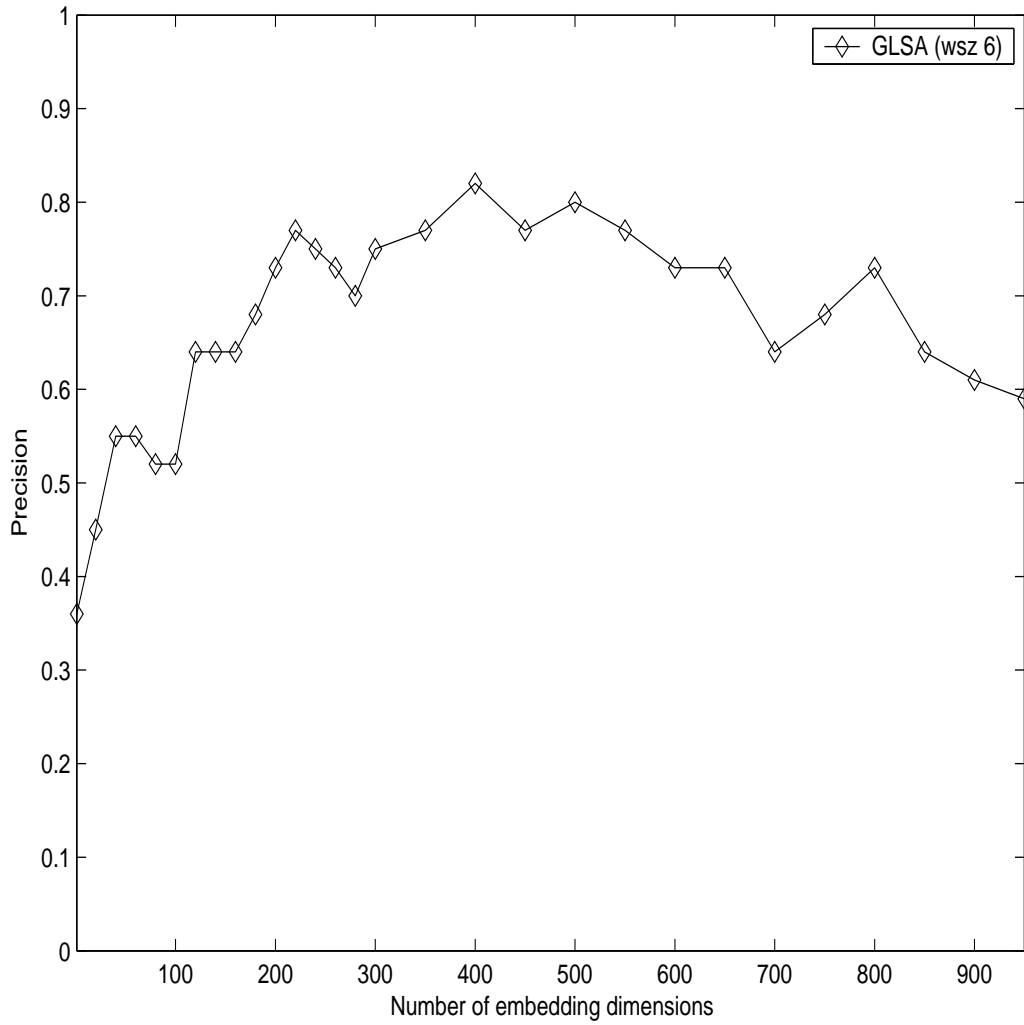


Figure 4: TS2 test. Measure = GLSA(pmi). Window size is 6. Precision at different numbers of GLSA dimensions.

| Window Size | PMI(T&C) | PMI  | GLSA |
|-------------|----------|------|------|
| 2           | 0.66     | 0.51 | 0.80 |
| 4           | 0.65     | 0.52 | 0.78 |
| 8           | 0.70     | 0.67 | 0.73 |
| 16          | 0.69     | 0.64 | 0.78 |
| 20          | NA       | 0.69 | 0.75 |
| 256         | 0.75     | NA   | NA   |

Table 2: Precision for the TS2 synonyms test. The maximum precision obtained by Terra and Clarke is 75% (with window size 256).

## 4.2 Text Classification

### 4.2.1 Document Collection

For the text classification experiments we used the WebKB data set containing 4518 web pages from different universities. We used the web pages from the largest classes: student, faculty, course, project. The vocabulary size for GLSA computations was 8224 terms (we used only the terms that occurred in min 5 documents), 567 terms from the WebKB collection did not occur in our English Gigaword corpus and were discarded.

### 4.2.2 Experimental Setup

- Index the WebKB documents with the index obtained from the English Gigaword collection
- Compute co-occurrence statistics for the WebKB vocabulary terms using the English Gigaword collection.
- Compute the  $GLSA\_PMI$  term vectors
- Compute new document vectors

We used the libsvm package to do the classification task on the WebKB data set. We report the classification accuracy computed with the leave-one-out procedure. In all cases we scaled the entries in the document vectors to be  $\in [-1, 1]$  and used the linear kernel that has been shown to perform well on text data.

We ran three sets of the classification experiments: using the GLSA-based document vectors, using document vectors indexed with

| Language model | $V_{all}$ | $V_{pruned}$ | GLSA |
|----------------|-----------|--------------|------|
| ML             | 0.70      | 0.76         | 0.84 |

Table 3: Classification accuracy on the WebKB data set for the one vs. all binary classification task. We assigned 1641 student pages to be one class, 2877 pages from other classes to be the other class. For GLSA the best performance over all embedding dimensions is reported.

| Language model | $V_{all}$ | $V_{pruned}$ | GLSA |
|----------------|-----------|--------------|------|
| ML             | 0.62      | 0.61         | 0.78 |
| Laplace        | 0.61      | 0.60         | 0.79 |
| Dirichlet      | 0.62      | 0.60         | 0.78 |

Table 4: F-score for the pairs test on the WebKB data set for three types of document vectors.

a particular language model using all vocabulary terms, and using document vectors indexed with a particular language model but omitting the vocabulary terms that were not used for the GLSA computations. We computed the term weights for the document vectors using the maximum likelihood (ML) language model. Table 3 shows the result for the binary classification task student pages vs web pages from all other classes. It can be seen that the accuracy is much higher when the GLSA vectors are used.

### 4.2.3 Pairs Test

As a further estimation of the quality of the GLSA term and document representation we performed the pairs test, following Platt [15]. For each of the 4199 labelled documents compute its nearest neighbour in the embedded space, in terms of the cosine measure. A is the number of such pairs that have the same labels, B is the number of pairs that have two different labels. The F-score is computed as:

$$F - \text{score} = A / (A + \frac{B}{2})$$

The motivation is that documents that belong to the same class are representation we performed the pairs test, following Platt [15] similar and therefore their vector representations should be similar as well.

Figure 5 shows the F-score with different numbers of embedding

dimensions and three different language models. Again, the GLSA term vectors improve the performance.

## 5 Conclusion

We presented an algorithmic framework of Generalized Latent Semantic Analysis for computing a low dimensional vector space representation for documents. This approach combines the insights from Web-based synonymy extraction and dimensionality reduction for information retrieval.

We consider document vectors to be a linear combination of low dimensional term vectors. A low dimensional vector space is computed using the point-wise mutual information or other corpus-based measures of semantic association between terms as similarity measure. Once we have pair-wise similarity information, we can apply MDS or any other dimensionality reduction method to obtain term vectors.

One of the advantages of GLSA over LSA is that linguistically well founded pair-wise similarity information for terms can be obtained without using term-document vectors explicitly. Consistent with other research we found that point-wise mutual information (PMI) is well suited for this application. Semantic relationship between terms is not dependent on a particular document collection, so we can use additional resources such as the Web to refine our PMI computations.

The GLSA term vectors were shown to improve performance on different tasks such as synonymy test, nearest neighbor test and document classification.

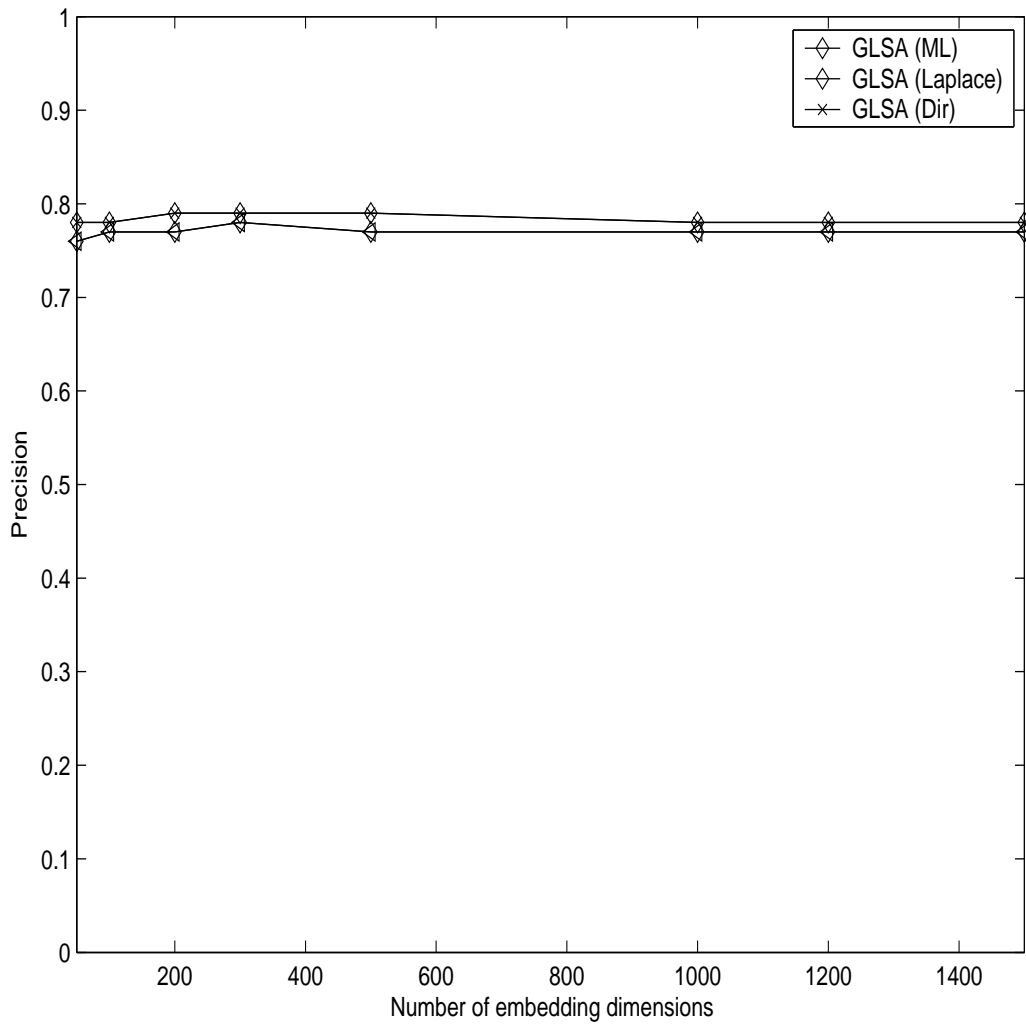


Figure 5: F-score for the pairs test for the WebKB data set at different numbers of embedding dimensions, using different language models to compute term weights.

## References

- [1] R. K. Ando. Latent semantic space: iterative scaling improves precision of inter-document similarity measurement. In *SIGIR '00: Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 216–223. ACM Press, 2000.
- [2] L. D. Baker and A. K. McCallum. Distributional clustering of words for text classification. In W. B. Croft, A. Moffat, C. J. van Rijsbergen, R. Wilkinson, and J. Zobel, editors, *Proceedings of SIGIR-98, 21st ACM International Conference on Research and Development in Information Retrieval*, pages 96–103, Melbourne, AU, 1998. ACM Press, New York, US.
- [3] R. Bekkerman, R. El-Yaniv, and N. Tishby. Distributional word clusters vs. words for text categorization.
- [4] M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15(6):1373–1396, 2003.
- [5] N. J. Belkin and W. B. Croft. Information filtering and information retrieval: two sides of the same coin? *Commun. ACM*, 35(12):29–38, 1992.
- [6] T. Chklovski and P. Pantel. Verbocean: Mining the web for fine-grained semantic verb relations. In *Proceedings of Conference on Empirical Methods in Natural Language Processing (EMNLP-04)*, 2004.
- [7] N. Cristianini, J. Shawe-Taylor, and H. Lodhi. Latent semantic kernels. *J. Intell. Inf. Syst.*, 18(2-3):127–152, 2002.
- [8] S. C. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, and R. A. Harshman. Indexing by latent semantic analysis. *Journal of the American Society of Information Science*, 41(6):391–407, 1990.
- [9] S. Dumais. Enhancing performance in latent semantic indexing. *TM-ARH-017527 Technical Report, Bellcore, 1990.*, 1990.
- [10] X. He and P. Niyogi. Locality preserving projections. In *NIPS*, 2003.
- [11] G.-A. Levow, D. W. Oard, and P. Resnik. Dictionary-based techniques for cross-language information retrieval. *(to appear) In-*

*formation Processing and Management: Special Issue on Cross-language Information Retrieval.*

- [12] C. Manning and H. Schuetze. *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, MA, 1999.
- [13] P. Pantel and D. Lin. Document clustering with committees. In *SIGIR '02: Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 199–206. ACM Press, 2002.
- [14] C. H. Papadimitriou, H. Tamaki, P. Raghavan, and S. Vempala. Latent semantic indexing: A probabilistic analysis. In *Proceedings of the ACM Conference on Principles of Database Systems (PODS), Seattle*, pages 159–168, 1998.
- [15] J. Platt. Fastmap, metricmap, and landmark mds are all nystrm algorithms. In *10th International Workshop on Artificial Intelligence and Statistics*, 2005.
- [16] M. Sahlgren and R. Coester. Using bag-of-concepts to improve the performance of support vector machines in text categorization. In *20th International Conference on Computational Linguistics, COLING*, pages 487–493, 2004.
- [17] G. Salton and M. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, 1983.
- [18] H. Schuetze. Automatic word sense discrimination. *Computational Linguistics*, 24(21):97–124, 1998.
- [19] N. Slonim and N. Tishby. Document clustering using word clusters via the information bottleneck method. In *Research and Development in Information Retrieval*, pages 208–215, 2000.
- [20] J. B. Tenenbaum, V. de Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.
- [21] E. L. Terra and C. L. A. Clarke. Frequency estimates for statistical word similarity measures. In *HLT-NAACL*, 2003.
- [22] P. D. Turney. Mining the web for synonyms: PMI-IR versus LSA on TOEFL. *Lecture Notes in Computer Science*, 2167:491–502, 2001.
- [23] D. Widdows. A mathematical model for context and word-meaning. In *CONTEXT*, pages 369–382, 2003.

- [24] D. Widdows. Unsupervised methods for developing taxonomies by combining syntactic and statistical information. In *HLT-NAACL*, 2003.