

Terms and Document Representation with Generalized Latent Semantic Analysis

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 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. We also show that term-based document representation improves performance on the document classification test.

1 Introduction

Document indexing and representation of term-document relations are crucial for document classification, clustering and retrieval (Salton and McGill, 1983; Ponte and Croft, 1998; Deerwester et al., 1990). 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 and McGill, 1983). In the traditional bag-of-words representation (Salton and McGill, 1983) of the document vector, words rep-

resent 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., 2003) and expanding the document and query vectors with synonyms and related terms as discussed in (Levow et al., 2005), improve the performance on average. However, they also introduce some instability and thus increased variance (Levow et al., 2005). The language modelling approach (Ponte and Croft, 1998; Berger and Lafferty, 1999) 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., 1990) is one of the best known dimensionality reduction algorithms. 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, 2000). Different related dimensionality reduction techniques proved successful for document clustering and retrieval (Belkin and Niyogi, 2003; He et al., 2004; Callan et al., 2003).

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. We begin with semantically motivated pair-wise term similarities to compute a term representation. This shift from dual document-term representation to term representation has the following motivation.

1. 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 has been shown to improve performance on such tasks as the synonymy test, taxonomy induction, document clustering (Turney, 2001; Terra and Clarke, 2003; Chklovski and Pantel, 2004; Widdows, 2003). On the other hand, many semi-supervised and transductive methods based on document vectors cannot yet handle such large document collections.
2. While the vocabulary size is still quite large, it is intuitively clear that the intrinsic dimensionality of the vocabulary space is much lower. Content bearing words, i.e. words which convey 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 and the Laplacian Eigenmaps Embedding algorithm, to compute GLSA term and document vectors. We show that the GLSA terms representation outperforms related approaches on the synonymy test. Furthermore, we use the cosine similarity between the GLSA term vectors as term translation probability within the language modelling approach. Improves performance on document classification. Our experiments also demonstrate that GLSA document vectors improve the per-

formance on the document classification task.

The rest of the paper is organized as follows. Section 2 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 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^1 , obtain a matrix of pair-wise similarities, S , using the large corpus W
3. Obtain the matrix U^T of 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.

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.

2.1.1 Document Vectors

GLSA does not have the out-of-sample problem for new documents. It does have this problem for new terms, but new terms appear in a much lower rate than documents. In addition, new rare terms will

¹It 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, this approach would suffer from noise introduced by typos and infrequent and non-informative words. Methods of efficient filtering of the core vocabulary are subject of future work.

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 its 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.

2.1.2 Term Translation Probabilities

The language modelling approach (Ponte and Croft, 1998; Berger and Lafferty, 1999) proved very effective for the information retrieval task. This method assumes that every document defines a multinomial probability distribution $p(w|d)$ over the vocabulary space. Thus, given a query $\mathbf{q} = (q_1, \dots, q_m)$, the likelihood of the query is estimated using the document’s distribution: $p(\mathbf{q}|d) = \prod_1^m p(q_i|d)$. Relevant documents maximize $p(d|q) \propto p(q|d)p(d)$. The probability $p(w|d)$ is estimated from a corpus usually using different types of smoothing methods. Berger and Lafferty (Berger and Lafferty, 1999) proposed to incorporate “semantic smoothing” into the likelihood estimation. Motivated by the statistical machine translation model they introduced translation probabilities between words into the document-to-query model. The intuition is that even if a document does not contain the same terms as the query, it may contain terms that are semantically related and thus have high probability of being “translations”, i.e. reformulations for the query words. Thus, they used

$$p(\mathbf{q}|d) = \prod_i \sum_{w \in d} t(q_i|w)p(w|d). \quad (1)$$

This approach showed improvements over the baseline language modelling approach (Berger and Lafferty, 1999). The estimation of the translation probabilities is, however, a difficult task. Lafferty and Zhai used a Markov chain on words and documents to estimate the translation probabilities (Lafferty and Zhai, 2001).

We propose to use the GLSA framework for inducing the translation probabilities between terms. We used appropriately normalized values of the

cosine similarity measure between pairs of terms $\cos(\vec{t}_i, \vec{t}_j)$ as the translation probability $t(t_i|t_j)$. For the document classification task, we computed the likelihood of a test document d given a training document d_i as

$$p(d|d_i) = \prod_{v \in d} \sum_{w \in d_i} t(v|w)p(w|d_i). \quad (2)$$

In addition, the inner products between the term vectors are implicitly used when the similarity between the document vectors is computed and can be interpreted as “translation” probabilities between the terms:

$$\langle \vec{d}_i, \vec{d}_j \rangle = \sum_{w \in d_i} \sum_{v \in d_j} \alpha_w^{d_i} \beta_v^{d_j} \langle \vec{w}, \vec{v} \rangle, \quad (3)$$

where $\alpha_w^{d_i}$ and $\beta_v^{d_j}$ represent term’s score with respect to the document. Our experiments show that these two methods offer an advantage for document classification.

2.2 Low-dimensional Representation

2.2.1 Singular Value Decomposition

Since S is a real symmetric matrix, it is diagonalizable, so that $S = U\Sigma U^T$. If it also positive semi-definite, it can be represented as a product of two matrices $S = \hat{U}\hat{U}^T$, so that $\hat{U} = U\Sigma^{1/2}$.

Eckart and Young (Cox and Cox, 2001) have shown that given any matrix S and its singular value decomposition (SVD) $S = U\Sigma V^T$, 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

$$S_k = \operatorname{argmin}_X \|S - X\|_F^2 \quad (4)$$

The SVD of a symmetric matrix of pair-wise term similarities S is the same as its eigenvalue decomposition. Therefore, the first 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.

LSA is one special case within the GLSA framework. LSA uses SVD to compute the rank k approximation to the matrix of pair-wise term similarities that are computed using the inner products between the term vectors in the space of documents (Bartell et al., 1992). 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., 1998).

2.2.2 Laplacian Eigenmaps Embedding

In our experiments we also used the Laplacian Eigenmaps Embedding algorithm (Belkin and Niyogi, 2003) to compute low-dimensional term vectors. Laplacian Eigenmaps Embedding assumes that the high dimensional data, such as images and documents, intrinsically belongs to a low dimensional manifold, and preserves the similarities in S only locally since local information is often more reliable. Since it does not require distance information for the input data, it is well suited for the GLSA framework.

The Laplacian Eigenmaps Embedding algorithm computes the low dimensional vectors y to minimize under certain constraints

$$\sum_{ij} \|y_i - y_j\|^2 W_{ij}. \quad (5)$$

W is the weight matrix based on the graph adjacency matrix. The solution to this problem are the generalized eigenvectors of $Ly = \lambda Zy$ corresponding to the k smallest non-zero eigenvalues. Z is the diagonal matrix of degrees, $Z_{ii} = \sum_{j=1}^n W_{ij}$, and L is the graph Laplacian $L = D - W$, see (Belkin and Niyogi, 2003) for details.

A further reason why we choose the Laplacian Embedding algorithm is that it works well in practice, when the weights matrix W is binary. This is important for our approach since the term co-occurrence information is often used to test the hypothesis of their independence, so that not the value itself, but the test whether it surpasses a certain threshold is important.

2.3 Measures 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 and Schütze, 1999). In this paper we mainly use the point-wise mutual information because it has been successfully applied to such

semantic proximity tests as the synonymy test (Turney, 2001; Terra and Clarke, 2003) and taxonomy induction (Widdows, 2003). It was also successfully used as a measure of term similarity to compute document clusters (Pantel and Lin, 2002), and to extract semantic relations between verbs (Chklovski and Pantel, 2004).

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

$$PMI(w_1, w_2) = \log \frac{P(W_1 = 1, W_2 = 1)}{P(W_1 = 1)P(W_2 = 1)}. \quad (6)$$

The similarity matrix S with pairs-wise co-occurrence based scores may not be positive semi-definite. Since such matrices work well in practice, one common approach is to use only the eigenvectors corresponding to the positive eigenvalues (Cox and Cox, 2001). We used this approach 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, 2000) tries to put more weight on documents from underrepresented clusters of document to improve the performance of LSA on heterogeneous collections. Random Indexing (Sahlgren and Coester, 2004) projects the document vectors on random low-dimensional vectors. The Locality Preserving Indexing (He et al., 2004) is a linear approximation of the Laplacian Eigenmaps Embedding algorithm. 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 (Landauer and Dumais, 1997). Document vectors for other collections are constructed as linear combinations of LSA term vectors. LSA uses only one particular measure of term similarity. The Word Space Model for word sense disambiguation developed by Schütze (Schütze, 1998) is another special case of GLSA which computes the term vectors directly. In-

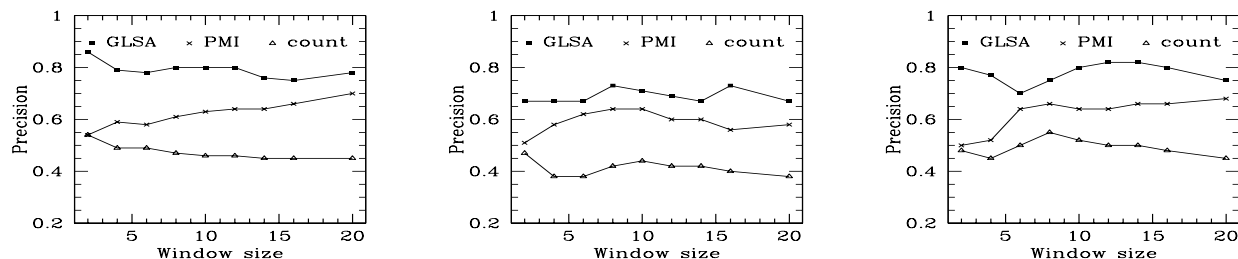


Figure 1: Precision with GLSA, PMI and count over different window sizes, for the TOEFL (left), TS1 (middle) and TS2 (right) synonymy tests.

stead 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, 1998; Widdows, 2003). Latent Relational Analysis (Turney, 2004) looks at pair-wise relations between terms and not at term vectors for the whole vocabulary and uses co-occurrence counts within context patterns.

4 Experiments

The goal of the experimental evaluation of the GLSA term vectors was twofold. First, we use the synonymy test to demonstrate that the GLSA vector space representation for terms captures their semantic relations. Our results demonstrate that similarities between GLSA term vectors achieve better results than PMI scores. Second, we performed document categorization tests to show that GLSA document vectors help to improve performance.

To collect the co-occurrence for the similarities matrix 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, our collection had 1,119,364 documents with 771,451 terms. We used the Lemur toolkit² to tokenize and index all document collections used in our experiments, we used stemming and a list of stop words. We denote the combination of the PMI similarities matrix with SVD as $GLSA$, and with the Laplacian Embedding as $GLSA_L$. Unless stated otherwise, for the GLSA methods we report the best performance over different numbers of embedding dimensions.

²<http://www.lemurproject.org/>

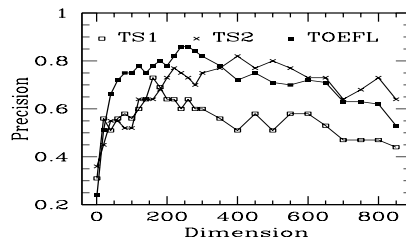


Figure 2: Precision on the synonymy tests at different numbers of GLSA dimensions with the best window size.

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 and Dumais, 1997). Turney (Turney, 2001) and (Terra and Clarke, 2003) showed that the PMI-IR approach achieves higher accuracy rates on this tests than LSA. The PMI-IR approach uses the PMI scores computed using the Web or a large Web based document collection to answer the synonymy questions. We evaluated the GLSA term vectors on the synonymy test and compared the results to (Terra and Clarke, 2003). Terra et al. (Terra and Clarke, 2003) compared the performance of different co-occurrence based measures of term similarity on the synonymy test and came to the conclusion that PMI yielded best results. Following (Terra and Clarke, 2003), we used the TOEFL, TS1 and TS2 synonymy tests. The TOEFL test contains 80 synonymy questions. Since GLSA in its present formulation cannot handle multi-word expressions, we had to modify the TS1 and TS2 tests slightly. We

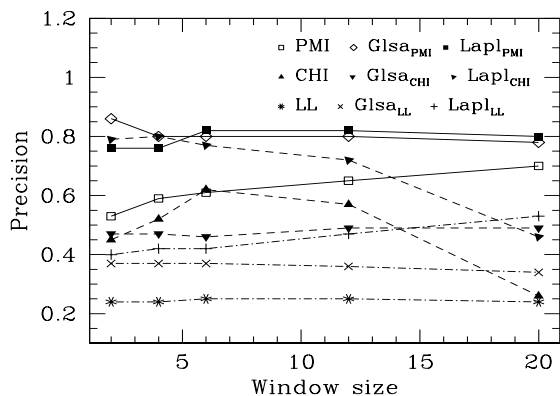


Figure 3: TOEFL test precision for PMI, χ^2 (CHI) and log likelihood ratio (LL) measures alone; with SVD (GLSA_{PMI}, GLSA_{CHI}, GLSA_{LL}) and with the Laplacian Embedding (Lapl_{PMI}, Lapl_{CHI}, Lapl_{LL}).

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 those in (Terra and Clarke, 2003) is only suggestive.

4.1.1 GLSA Setting

To have a richer vocabulary space, we added 2000 most frequent words from the English Gigaword collections to the vocabularies of the TOEFL, TS1 and TS2 tests. We computed separate GLSA term vectors for the extended vocabularies of these synonymy tests and selected the term whose term vector had the highest cosine similarity to the question term vector as the synonym. We computed precision 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, 1998; Terra and Clarke, 2003). Since the performance of co-occurrence based measures is sensitive to the window size, we report the results for different window sizes. For the TOEFL set we combined PMI, χ^2 and log likelihood ratio similarities with SVD and the Laplacian Embedding.

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, 2003). One explanation for this discrepancy is the size and the composition of the document collections used for the co-occurrence statistics. Our collection is smaller and, more importantly, less heterogeneous than the web based collection in (Terra and Clarke, 2003). 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 and Clarke, 2003). GLSA achieves the same maximum precision as in (Terra and Clarke, 2003) for TS1 (0.73) and higher precision on TS2 (0.82 vs. 0.75 in (Terra and Clarke, 2003)).

Figure 2 shows the precision for the GLSA terms only, using different number of dimensions. The variation of precision at different numbers of embedding dimensions is somewhat high for TS1 but much smoother for the TOEFL and TS2 tests.

As Figure 3 shows, we obtained rather poor results with using the χ^2 and log likelihood measures alone and in combination with SVD. As mentioned above, however, for these test, the similarity values themselves may not be well suited for this task. Since the Laplacian Embedding only uses the ordering information to construct the graph, we were particularly interested in combining these measures with the Laplacian Embedding algorithm. The combination of PMI with the Laplacian Embedding outperformed the baseline and had the highest precision somewhat below the PMI with SVD combination. Although GLSA_L improved the results when combined with the log likelihood ratio test, it still did not perform well compared to other measures. However, the Laplacian Embedding significantly improved the performance when the χ^2 test was used.

4.2 Document Classification

We conducted document classification experiments to demonstrate the advantage of the GLSA docu-

#L	$Group_d$			$Group_s$		
	tf	Glsa	LM	tf	Glsa	LM
100	0.58	0.75	0.69	0.42	0.48	0.48
200	0.65	0.78	0.74	0.47	0.52	0.51
400	0.69	0.79	0.76	0.51	0.56	0.55
1000	0.75	0.81	0.80	0.58	0.60	0.59
2000	0.78	0.83	0.83	0.63	0.64	0.63

Table 1: k -NN classification accuracy for 20NG.

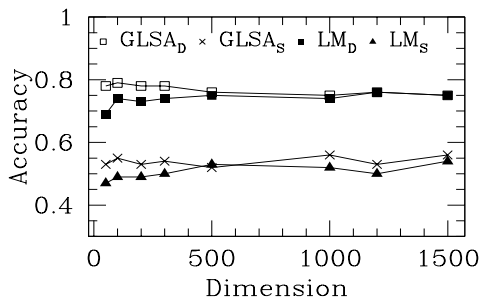


Figure 4: k -NN with 400 training documents.

ment vectors over the bag-of-words representation. We also used the GLSA-based term translation probabilities within the language modelling framework.

We used the 20 news groups data set because previous studies showed that the classification performance on this document collection can noticeably benefit from additional semantic information (Bekkerman et al., 2003). For the GLSA computations we used the terms that occurred in at least 15 documents, and had a vocabulary of 9732 terms. We removed documents with fewer than 5 words. For the experiments reported here, we used 2 sets of 6 news groups. $Group_d$ contained documents from dissimilar news groups³, with a total of 5300 documents. $Group_s$ contained documents from more similar news groups⁴ and had 4578 documents.

We ran the k -NN classifier with $k=5$ on ten random splits of training and test sets, with different numbers of training documents. The baseline was to use the cosine similarity between the bag-of-words document vectors weighted with term frequency. We used some other weighting schemes

³os.ms, sports.baseball, rec.autos, sci.space, misc.forsale, religion-christian

⁴politics.misc, politics.mideast, politics.guns, religion.misc, religion.christian, atheism

such as maximum likelihood and Laplace smoothing, but they did not improve the results.

We computed the score between the training and test documents using two approaches: cosine similarity between the document vectors according to Equation 3 (denoted as $GLSA$), and the language modelling score which included the translation probabilities between the terms as in Equation 2 (denoted as LM). We used the term frequency as an estimate for $p(w|d)$. To compute the matrix of translation probabilities P , where $P[i][j] = t(t_i|t_j)$ for the LM_{GLSA} approach, we first obtained the matrix $\hat{P}[i][j] = \cos(\vec{t}_i, \vec{t}_j)$. To be able to construct a probability distribution $t(t_i|t_j)$ we set the negative and zero entries to a small positive value. Finally, we normalized the rows of \hat{P} to sum up to one.

Table 1 shows that for both settings GLSA and LM outperform the tf document vectors. As expected, the classification task was more difficult for the similar news groups. However, in this case both GLSA-based approaches outperform the baseline. In both cases, the advantage is more significant with smaller sizes of the training set. GLSA and LM performance peaked at around 300-500 dimensions which is in line with results for other SVD-based approaches (Deerwester et al., 1990; He et al., 2004). In cases, when the highest accuracy was achieved at higher dimensions, the increase after 500 dimensions was rather small, as illustrated in Figure 4.

These results illustrate that the pair-wise similarities between the GLSA term vectors add important semantic information which helps to go beyond term matching and deal with synonymy and polysemy.

5 Conclusion and Future Work

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. Once we have a set of most informative words, we can use approximation methods for the eigenvalue decomposition.

We have presented the GLSA framework for computing semantically motivated term and document vectors. This framework allows us to take advan-

tage 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 eigenvalue decomposition we have obtained term vectors that outperform the state-of-the-art approaches on the synonymy test. We have also showed that the term-based document representation and GLSA-based term translation probabilities improve performance on the document classification task.

References

- Rie Kubota Ando. 2000. Latent semantic space: iterative scaling improves precision of inter-document similarity measurement. In *Proc. of the 23rd ACM SIGIR*, pages 216–223.
- Brian T. Bartell, Garrison W. Cottrell, and Richard K. Belew. 1992. Latent semantic indexing is an optimal special case of multidimensional scaling. In *Proc. of the 15th ACM SIGIR*, pages 161–167. ACM Press.
- Ron Bekkerman, Ran El-Yaniv, and Naftali Tishby. 2003. Distributional word clusters vs. words for text categorization.
- Mikhail Belkin and Partha Niyogi. 2003. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15(6):1373–1396.
- Adam Berger and John Lafferty. 1999. Information retrieval as statistical translation. In *Proc. of the 22rd ACM SIGIR*.
- Jamie Callan, Gordon Cormack, Charles Clarke, David Hawking, and Alan Smeaton. 2003. Document clustering based on non-negative matrix factorization. In *Proc. of the 26rd ACM SIGIR*, New York. ACM.
- Timothy Chklovski and Patrick Pantel. 2004. Verbocean: Mining the web for fine-grained semantic verb relations. In *Proc. of EMNLP*.
- T. Cox and M. Cox. 2001. *Multidimensional Scaling*. CRC/Chapman and Hall.
- Scott C. Deerwester, Susan T. Dumais, Thomas K. Landauer, George W. Furnas, and Richard A. Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American Society of Information Science*, 41(6):391–407.
- Xiaofei He, Deng Cai, Haifeng Liu, and Wei-Ying Ma. 2004. Locality preserving indexing for document representation. In *Proc. of the 27rd ACM SIGIR*, pages 96–103. ACM Press.
- John Lafferty and Chengxiang Zhai. 2001. Document language models, query models, and risk minimization for information retrieval. In *Proc. of the 24th ACM SIGIR*, pages 111–119, New York, NY, USA. ACM Press.
- Thomas K. Landauer and Susan T. Dumais. 1997. A solution to platos problem: The latent semantic analysis theory of the acquisition, induction, and representation of knowledge. *Psychological Review*.
- Gina-Anne Levow, Douglas W. Oard, and Philip Resnik. 2005. Dictionary-based techniques for cross-language information retrieval. *Information Processing and Management: Special Issue on Cross-language Information Retrieval*.
- Chris Manning and Hinrich Schütze. 1999. *Foundations of Statistical Natural Language Processing*. MIT Press. Cambridge, MA.
- Patrick Pantel and Dekang Lin. 2002. Document clustering with committees. In *Proc. of the 25th ACM SIGIR*, pages 199–206. ACM Press.
- Jay M. Ponte and W. Bruce Croft. 1998. A language modeling approach to information retrieval. In *Proc. of the 21st ACM SIGIR*, pages 275–281, New York, NY, USA. ACM Press.
- M Sahlgren and R Coester. 2004. Using bag-of-concepts to improve the performance of support vector machines in text categorization. In *Proc. of the 20th COLING*, pages 487–493.
- G Salton and M McGill. 1983. *Introduction to Modern Information Retrieval*. McGraw-Hill.
- B. Schölkopf, A.J. Smola, and K.-R. Muller. 1998. Non-linear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10.
- Hinrich Schütze. 1998. Automatic word sense discrimination. *Computational Linguistics*, 24(21):97–124.
- Egidio L. Terra and Charles L. A. Clarke. 2003. Frequency estimates for statistical word similarity measures. In *Proc. of HLT-NAACL*.
- Peter D. Turney. 2001. Mining the web for synonyms: PMI-IR versus LSA on TOEFL. *Lecture Notes in Computer Science*, 2167:491–502.
- Peter D. Turney. 2004. Human-level performance on word analogy questions by latent relational analysis. Technical report, Technical Report ERB-1118, NRC-47422.
- Dominic Widdows. 2003. Unsupervised methods for developing taxonomies by combining syntactic and statistical information. In *Proc. of HLT-NAACL*.