# Graph-based Generalized Latent Semantic Analysis for Document Representation

**Irina Matveeva**
Dept. of Computer Science
University of Chicago
Chicago, IL 60637
matveeva@cs.uchicago.edu

**Gina-Anne Levow**
Dept. of Computer Science
University of Chicago
Chicago, IL 60637
levow@cs.uchicago.edu

## Abstract

Document indexing and representation of term-document relations are very important for document clustering and retrieval. In this paper, we combine a graph-based dimensionality reduction method with a corpus-based association measure within the Generalized Latent Semantic Analysis framework. We evaluate the graph-based GLSA on the document clustering task.

## 1 Introduction

Document indexing and representation of term-document relations are very important issues for document clustering and retrieval. Although the vocabulary space is very large, content bearing words are often combined into semantic classes that contain synonyms and semantically related words. Hence there has been a considerable interest in low-dimensional term and document representations.

Latent Semantic Analysis (LSA) (Deerwester et al., 1990) is one of the best known dimensionality reduction algorithms. The dimensions of the LSA vector space can be interpreted as latent semantic concepts. The cosine similarity between the LSA document vectors corresponds to documents' similarity in the input space. LSA preserves the documents similarities which are based on the inner products of the input bag-of-word documents and it preserves these similarities globally.

More recently, a number of graph-based dimensionality reduction techniques were successfully applied to document clustering and retrieval (Belkin and Niyogi, 2003; He et al., 2004). The main advantage of the graph-based approaches over LSA is the notion of locality. Laplacian Eigenmaps Embedding (Belkin and Niyogi, 2003) and Locality Preserving Indexing (LPI) (He et al., 2004) discover the local structure of the term and document space and compute a semantic subspace with a stronger discriminative power. Laplacian Eigenmaps Embedding and LPI preserve the input similarities only locally, because this information is most reliable. Laplacian Eigenmaps Embedding does not provide a fold-in procedure for unseen documents. LPI is a linear approximation to Laplacian Eigenmaps Embedding that eliminates this problem. Similar to LSA, the input similarities to LPI are based on the inner products of the bag-of-word documents. Laplacian Eigenmaps Embedding can use any kind of similarity in the original space.

Generalized Latent Semantic Analysis (GLSA) (Matveeva et al., 2005) is a framework for computing semantically motivated term and document vectors. It extends the LSA approach by focusing on term vectors instead of the dual document-term representation. GLSA requires a measure of semantic association between terms and a method of dimensionality reduction.

In this paper, we use GLSA with point-wise mutual information as a term association measure. We introduce the notion of locality into this framework and propose to use Laplacian Eigenmaps Embedding as a dimensionality reduction algorithm. We evaluate the importance of locality for document representation in document clustering experiments.

The rest of the paper is organized as follows. Sec-



*Workshop on TextGraphs, at HLT-NAACL 2006*, pages 61–64,
New York City, June 2006. 

tion 2 contains the outline of the graph-based GLSA algorithm. Section 3 presents our experiments, followed by conclusion in section 4.

## 2 Graph-based GLSA

### 2.1 GLSA Framework

The GLSA algorithm (Matveeva et al., 2005) has the following setup. The input is a document collection $C$ with vocabulary $V$ and a large corpus $W$.

1. For the vocabulary in $V$, obtain a matrix of pair-wise similarities, $S$, using the corpus $W$
2. Obtain the matrix $U^T$ of a low dimensional vector space representation of terms that preserves the similarities in $S$, $U^T \in R^{k \times |V|}$
3. Construct the term document matrix $D$ for $C$
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 inner product between the term and document vectors in the GLSA space preserves the semantic association in the input space. The traditional term-document matrix is used in the last step to provide the weights in the linear combination of term vectors. LSA is a special case of GLSA that uses inner product in step 1 and singular value decomposition in step 2, see (Bartell et al., 1992).

### 2.2 Singular Value Decomposition

Given any matrix $S$, its singular value decomposition (SVD) is $S = U\Sigma V^T$. The matrix $S_k = U\Sigma_k V^T$ is obtained by setting all but the first $k$ diagonal elements in $\Sigma$ to zero. If $S$ is symmetric, as in the GLSA case, $U = V$ and $S_k = U\Sigma_k U^T$. The inner product between the GLSA term vectors computed as $U\Sigma_k^{1/2}$ optimally preserves the similarities in $S$ wrt square loss.

The basic GLSA computes the SVD of $S$ and uses $k$ eigenvectors corresponding to the largest eigenvalues as a representation for term vectors. We will refer to this approach as *GLSA*. As for LSA, the similarities are preserved globally.

### 2.3 Laplacian Eigenmaps Embedding

We used the Laplacian Embedding algorithm (Belkin and Niyogi, 2003) in step 2 of the GLSA algorithm to compute low-dimensional term vectors. Laplacian Eigenmaps Embedding preserves the similarities in $S$ only locally since local information is often more reliable. We will refer to this variant of GLSA as *GLSA*$_{\mathrm{L}}$.

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

$$\sum_{ij} ||y_i - y_j||^2 W_{ij}.$$

$W$ is the weight matrix based on the graph adjacency matrix. $W_{ij}$ is large if terms $i$ and $j$ are similar according to $S$. $W_{ij}$ can be interpreted as the penalty of mapping similar terms far apart in the Laplacian Embedding space, see (Belkin and Niyogi, 2003) for details. In our experiments we used a binary adjacency matrix $W$. $W_{ij} = 1$ if terms $i$ and $j$ are among the k nearest neighbors of each other and is zero otherwise.

### 2.4 Measure of Semantic Association

Following (Matveeva et al., 2005), we primarily used point-wise mutual information (PMI) as a measures of semantic association in step 1 of GLSA. PMI between random variables representing two words, $w_1$ and $w_2$, is computed as

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

### 2.5 GLSA Space

GLSA offers a greater flexibility in exploring the notion of semantic relatedness between terms. In our preliminary experiments, we obtained the matrix of semantic associations in step 1 of GLSA using point-wise mutual information (PMI), likelihood ratio and $\chi^2$ test. Although PMI showed the best performance, other measures are particularly interesting in combination with the Laplacian Embedding.

Related approaches, such as LSA, the Word Space Model (WS) (Schütze, 1998) and Latent Relational Analysis (LRA) (Turney, 2004) are limited to only one measure of semantic association and preserve the similarities globally.

Assuming that the vocabulary space has some underlying low dimensional semantic manifold. Laplacian Embedding algorithm tries to approximate this manifold by relying only on the local similarity information. It uses the nearest neighbors graph constructed using the pair-wise term similarities. The computations of the Laplacian Embedding uses the graph adjacency matrix $W$. This matrix can be binary or use weighted similarities. The advantage of the binary adjacency matrix is that it conveys the neighborhood information without relying on individual similarity values. It is important for co-occurrence based similarity measures, see discussion in (Manning and Schütze, 1999).

The Locality Preserving Indexing (He et al., 2004) has a similar notion of locality but has to use bag-of-words document vectors.

## 3 Document Clustering Experiments

We conducted a document clustering experiment for the Reuters-21578 collection. To collect the co-occurrence statistics for the similarities matrix $S$ we used a subset of the English Gigaword collection (LDC), containing New York Times articles labeled as "story". We had 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, with stemming and a list of stop words.

Since Locality Preserving Indexing algorithm (LPI) is most related to the graph-based GLSA$_L$, we ran experiments similar to those reported in (He et al., 2004). We computed the GLSA document vectors for the 20 largest categories from the Reuters-21578 document collection. We had 8564 documents and 7173 terms. We used the same list of 30 TREC words as in (He et al., 2004) which are listed in table 1[2]. For each word on this list, we generated a cluster as a subset of Reuters documents that contained this word. Clusters are not disjoint and contain documents from different Reuters categories.

We computed *GLSA*, *GLSA*$_{\mathrm{L}}$, LSA and LPI representations. We report the results for $k = 5$ for the $k$ nearest neighbors graph for LPI and Laplacian Embedding, and binary weights for the adjacency matrix. We report results for 300 embedding dimensions for GLSA, LPI and LSA and 500 dimensions for *GLSA*$_{\mathrm{L}}$.

We evaluate these representations in terms of how well the cosine similarity between the document vectors within each cluster corresponds to the true semantic similarity. We expect documents from the same Reuters category to have higher similarity.

For each cluster we computed all pair-wise document similarities. All pair-wise similarities were sorted in decreasing order. The term "inter-pair" describes a pair of documents that have the same label. For the k$^{th}$ inter-pair, we computed precision at $k$ as:

$$\mathrm{precision(p_k)} = \frac{\#\mathrm{inter-pairs}\, p_j, \mathrm{s.t.}\ j < k}{k},$$

where $p_j$ refers to the $j^{th}$ inter-pair. The average of the precision values for each of the inter-pairs was used as the average precision for the particular document cluster.

Table 1 summarizes the results. The first column shows the words according to which document clusters were generated and the entropy of the category distribution within that cluster. The baseline was to use the *tf* document vectors. We report results for *GLSA*, *GLSA*$_{\mathrm{L}}$, LSA and LPI. The LSA and LPI computations were based solely on the Reuters collection. For *GLSA* and *GLSA*$_{\mathrm{L}}$we used the term associations computed for the Gigaword collection, as described above. Therefore, the similarities that are preserved are quite different. For LSA and LPI they reflect the term distribution specific for the Reuters collection whereas for GLSA they are more general. By paired 2-tailed t-test, at p $\leq$ 0.05, *GLSA* outperformed all other approaches. There was no significant difference in performance of *GLSA*$_{\mathrm{L}}$, LSA and the baseline. Disappointingly, we could not achieve good performance with LPI. Its performance varies over clusters similar to that of other approaches but the average is significantly lower. We would like to stress that the comparison of our results to those presented in (He et al., 2004) are only suggestive since (He et al., 2004) applied LPI to each cluster separately and used PCA as preprocessing. We computed the LPI representation for the full collection and did not use PCA.

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

[2] We used 28 words because we used stemming whereas (He et al., 2004) did not, so that in two cases, two words were reduces to the same stem.

| word | tf | glsa | glsaL | lsa | lpi |
|---|---|---|---|---|---|
| agreement(1) | 0.74 | 0.73 | 0.73 | 0.75 | 0.46 |
| american(0.8) | 0.63 | 0.72 | 0.59 | 0.64 | 0.36 |
| bank(1.4) | 0.45 | 0.52 | 0.40 | 0.48 | 0.28 |
| control(0.7) | 0.78 | 0.82 | 0.80 | 0.80 | 0.58 |
| domestic(0.8) | 0.64 | 0.68 | 0.66 | 0.68 | 0.35 |
| export(0.8) | 0.64 | 0.65 | 0.70 | 0.67 | 0.37 |
| five(1.3) | 0.74 | 0.77 | 0.71 | 0.70 | 0.40 |
| foreign(1.2) | 0.51 | 0.58 | 0.55 | 0.56 | 0.28 |
| growth(1) | 0.51 | 0.58 | 0.48 | 0.54 | 0.32 |
| income(0.5) | 0.84 | 0.86 | 0.83 | 0.80 | 0.69 |
| increase(1.3) | 0.51 | 0.61 | 0.53 | 0.53 | 0.29 |
| industrial(1.2) | 0.59 | 0.66 | 0.58 | 0.61 | 0.34 |
| internat.(1.1) | 0.58 | 0.59 | 0.54 | 0.61 | 0.34 |
| investment(1) | 0.68 | 0.77 | 0.70 | 0.72 | 0.46 |
| loss(0.3) | 0.98 | 0.99 | 0.98 | 0.98 | 0.88 |
| money(1.1) | 0.70 | 0.62 | 0.71 | 0.65 | 0.38 |
| national(1.3) | 0.49 | 0.58 | 0.49 | 0.55 | 0.27 |
| price(1.2) | 0.53 | 0.63 | 0.57 | 0.57 | 0.29 |
| production(1) | 0.56 | 0.66 | 0.58 | 0.59 | 0.29 |
| public(1.2) | 0.58 | 0.60 | 0.57 | 0.57 | 0.31 |
| rate(1.1) | 0.61 | 0.62 | 0.64 | 0.60 | 0.35 |
| report(1.2) | 0.66 | 0.72 | 0.62 | 0.65 | 0.35 |
| service(0.9) | 0.59 | 0.66 | 0.56 | 0.61 | 0.39 |
| source(1.2) | 0.56 | 0.54 | 0.59 | 0.60 | 0.27 |
| talk(0.9) | 0.74 | 0.67 | 0.73 | 0.74 | 0.39 |
| tax(0.7) | 0.91 | 0.93 | 0.90 | 0.89 | 0.67 |
| trade(1) | 0.85 | 0.74 | 0.82 | 0.60 | 0.33 |
| world(1.1) | 0.63 | 0.65 | 0.68 | 0.66 | 0.33 |
| Av. Acc | 0.65 | 0.68 | 0.65 | 0.66 | 0.40 |

Table 1: Average inter-pairs accuracy.

The inter-pair accuracy depended on the categories distribution within clusters. For more homogeneous clusters, e.g. “loss”, all methods (except LPI) achieve similar precision. For less homogeneous clusters, e.g. “national”, ”industrial”, ”bank”, GLSA and LSA outperformed the *tf* document vectors more significantly.

## 4 Conclusion and Future Work

We introduced a graph-based method of dimensionality reduction into the GLSA framework. Laplacian Eigenmaps Embedding preserves the similarities only locally, thus providing a potentially better approximation to the low dimensional semantic space. We explored the role of locality in the GLSA representation and used binary adjacency matrix as similarity which was preserved and compared it to GLSA with unnormalized PMI scores.

Our results did not show an advantage of $GLSA_{\mathrm{L}}$. $GLSA_{\mathrm{L}}$ and LPI seem to be very sensitive to the parameters of the neighborhood graph. We tried different parameter settings but more experiments are required for a thorough analysis. We are also planning to use a different document collection to eliminate the possible effect of the specific term distribution in the Reuters collection. Further experiments are needed to make conclusions about the geometry of the vocabulary space and the appropriateness of these methods for term and document embedding.

## References


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.

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

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.

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

Irina Matveeva, Gina-Anne Levow, Ayman Farahat, and Christian Royer. 2005. Generalized latent semantic analysis for term representation. In *Proc. of RANLP*.

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

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