Hybrid Document Indexing with Spectral Embedding

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

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

Abstract

Document representation has a large impact on the performance of document retrieval and clustering algorithms. We propose a hybrid document indexing scheme that combines the traditional bag-of-words representation with spectral embedding. This method accounts for the specifics of the document collection and also uses semantic similarity information based on a large scale statistical analysis. Clustering experiments showed improvements over the traditional *tf-idf* representation and over the spectral methods based solely on the document collection.

1 Introduction

Capturing semantic relations between words in a document representation is a difficult problem. Different approaches tried to overcome the term independence assumption of the bag-of-words representation (Salton and McGill, 1983) for example by using distributional term clusters (Slonim and Tishby, 2000) and expanding the document vectors with synonyms, see (Levow et al., 2005). Since content words can be combined into semantic classes 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. In the LSA space documents are indexed with latent semantic concepts. LSA showed large performance improvements over the traditional *tf-idf* representation on small document collections (Deerwester et al., 1990) but often does not perform well on large heterogeneous collections.

LSA maps all words to low dimensional vectors. However, the notion of semantic relatedness is defined differently for subsets of the vocabulary. In addition, the numerical information, abbreviations and the documents' style may be very good indicators of their topic. However, this information is no longer available after the dimensionality reduction.

We use a hybrid approach to document indexing to address these issues. We keep the notion of latent semantic concepts and also try to preserve the specifics of the document collection. We use a low-dimensional representation only for nouns and represent the rest of the document's content as *tf-idf* vectors.

The rest of the paper is organized as follows. Section 2 discusses our approach. Section 3 reports the experimental results. We conclude in section 4.

2 Hybrid Document Indexing

This section gives the general idea of our approach. We divide the vocabulary into two sets: nouns and the rest of the vocabulary. We use a method of spectral embedding, as described below and compute a low-dimensional representation for documents using only the nouns. We also compute a *tf-idf* representation for documents using the other set of words. Since we can treat each latent semantic concept in the low-dimensional representation as part of the vocabulary, we combine the two vector representations for each document by concatenating them.

Proceedings of NAACL HLT 2007, Companion Volume, pages 113–116,
Rochester, NY, April 2007. ©2007 Association for Computational Linguistics

2.1 Spectral Embedding

Spectral methods comprise a family of algorithms that use a matrix of pair-wise similarities S and perform its spectral analysis, such as the eigenvalue decomposition, to embed terms and documents in a low-dimensional vector space. $S = U\Sigma U^T$, where the columns of U are its eigenvectors and Σ is a diagonal matrix with the eigenvalues.

If we have a matrix of pair-wise word similarities S, its first k eigenvectors U_k will be used to represent the words in the latent semantic space. Semantically related words will have high association with the same latent concepts and their corresponding vectors will be similar. Moreover, the vector similarity between the word vectors will optimally preserve the original similarities (Cox and Cox, 2001).

We use two approaches to compute spectral embedding for nouns. Latent Semantic Analysis (LSA) (Deerwester et al., 1990) and Generalized Latent Semantic Analysis (GLSA) (Matveeva et al., 2005). For both we used the eigenvalue decomposition as the embedding step. The difference is in the similarities matrix which we are trying to preserve.

2.2 Distributional Term Similarity

LSA and GLSA begin with a matrix of pair-wise term similarities S, compute its eigenvectors U and use the first k of them to represent terms and documents, for details see (Deerwester et al., 1990; Matveeva et al., 2005). The main difference in our implementation of these algorithms is the matrix of pair-wise word similarities. Since our representation will try to preserve them it is important to have a matrix of similarities which is linguistically motivated.

LSA uses the matrix of pair-wise similarities which is based on document vectors. For two words w_i and w_j in the document collection containing n documents d_k, the similarity is computed as

$$S(w_i, w_j) = \sum_{k=1:n} \mathrm{tf}(w_i, d_k)\mathrm{idf}(w_i) * \mathrm{tf}(w_j, d_k)\mathrm{idf}(w_j),$$

where $\mathrm{tf}(w_i, d_k)$ is the term frequency for w_i in d_k and $\mathrm{idf}(w_i)$ is the inverse document frequency weight for w_i. LSA is a special case of spectral embedding restricted to one type of term similarities and dimensionality reduction method.

GLSA (Matveeva et al., 2005) generalizes the idea of latent semantic space. It proposes to use different types of similarity matrix and spectral embedding methods to compute a latent space which is closer to true semantic similarities. One way to do so is to use a more appropriate similarities matrix S.

PMI We use point-wise mutual information (PMI) to compute the matrix S. PMI between random variables representing the words w_i and w_j is computed as

$$PMI(w_i, w_j) = \log \frac{P(W_i = 1, W_j = 1)}{P(W_i = 1)P(W_j = 1)}.$$

Thus, for GLSA, $S(w_i, w_j) = PMI(w_i, w_j)$.

Co-occurrence Proximity An advantage of PMI is the notion of proximity. The co-occurrence statistics for PMI are typically computed using a sliding window. Thus, PMI will be large only for words that co-occur within a small fixed context. Our experiments show that this is a better approximation to true semantic similarities.

2.3 Document Indexing

We have two sets of the vocabulary terms: a set of nouns, N, and the other words, T. We compute *tf-idf* document vectors indexed with the words in T:

$$\vec{d_i} = (\alpha_i(w_1), \alpha_i(w_2), ..., \alpha_i(w_{|T|})),$$

where $\alpha_i(w_t) = \mathrm{tf}(w_t, d_i) * \mathrm{idf}(w_t)$.

We also compute a k-dimensional representation with latent concepts c_i as a weighted linear combination of LSA or GLSA term vectors $\vec{w_t}$:

$$\vec{d_i} = (c_1, ..., c_k) = \sum_{t=1:|T|} \alpha_i(w_t) * \vec{w_t},$$

We concatenate these two representations to generate a hybrid indexing of documents:

$$\vec{d_i} = (\alpha_i(w_1), ..., \alpha_i(w_{|T|}), c_1, ...c_k)$$

3 Experiments

We performed document clustering experiments to validate our approach.

Subset m-n	#topics	min #d	max #d	av. #d
5-10	19	6	10	8.2
50-150	21	55	150	94.7
500-1000	2	544	844	694.0
1000-5000	3	1367	2083	1792.3

Table 1: TDT2 topic subsets containing between m and n documents: the number of topics per subset, the minimum, the maximum and the average number of documents per topic in each subset.

Indexing		
All words	Nouns	Hybrid
tf-idf, LSA GLSA, $GLSA\_local$	*tf-idf*$_N$ GLSA$_N$	 *tf-idf*+GLSA$_N$

Table 2: Indexing schemes: with full vocabulary (All), only nouns (Nouns) and the combination.

Data We used the TDT2 collection[1] of news articles from six news agencies in 1998. We used only 10,329 documents that are assigned to one topic. TDT2 documents are distributed over topics very unevenly. We used subsets of the TDT2 topics that contain between m and n documents, see Table 1. We used the Lemur toolkit[2] with stemming and stop words list for the *tf-idf* indexing, Bikel's parser[3] to obtain the set of nouns and the PLAPACK package (Bientinesi et al., 2003) to compute the eigenvalue decomposition.

Global vs. Local Similarity To obtain the PMI values for GLSA we used the TDT2 collection, denoted as $GLSA_{local}$. Since co-occurrence statistics based on larger collections gives a better approximation to linguistic similarities, we also used 700,000 documents from the English GigaWord collection, denoted as GLSA and GLSA$_N$. We used a window of size 8.

Representations For each document we computed 7 representations, see Table 2. The vocabulary size we used with the *tf-idf* indexing was 114,127. For computational reasons we used the set of words that occurred in at least 20 documents with our spectral methods. We used 17,633 words for indexing with LSA and $GLSA_{local}$ and 17,572 words for GLSA. We also indexed documents using only the 15,325 nouns: *tf-idf*$_N$ and $GLSA_N$. The hybrid representation was computed using the *tf-idf* indexing without nouns and the $GLSA_N$ nouns vectors.

Evaluation We used the minimum squared residue co-clustering algorithm [4]. We report two evaluation measures: accuracy and the F1-score. The clustering algorithm assigns each document to a cluster. We map the cluster id's to topic labels using the Munkres assignment algorithm (Munkres, 1957) and compute the accuracy as the ratio of the correctly assigned labels.

The F1 score for cluster c_i labeled with topic t_i is computed using $F1 = \frac{2(p*r)}{(p+r)}$ where p is precision and r is recall. For clusters $C = (c_1, ..., c_n)$ and topics $T = (t_1, ..., t_n)$ we compute the total score:

$$F1(C,T) = \sum_{t \in T} \frac{N_t}{N} \max_{c \in C} F1(c,t).$$

N_t is the number of documents belonging to the topic t and N is the total number of documents. This measure accounts for the topic size and also corrects the topic assignments to clusters by using the max.

4 Results and Conclusion

Table 3 shows that the spectral methods outperform the *tf-idf* representations and have smaller variance. We report the performance for four subsets. The subset $5-10$ has a large number of topics, each with a similar number of documents. The subset $50-150$ has a large number of topics with a less even distribution of documents. $500-1000$ and $1000-5000$ have a couple of large topics. We ran the clustering over 30 random initializations. To eliminate the effect of the initial conditions on the performance we also used one document per cluster to seed the initial assignment for the $5-10$ subset.

All methods have the worst performance for the $5-10$ subset. The best performance is for the subset $500-1000$. LSA and $GLSA_{local}$ indexing are computed based on the TDT2 collection. $GLSA_{local}$ has better average performance which confirms that the co-occurrence proximity is important for distributional similarity. The GLSA indexing computed using a large corpus performs significantly worse than

[1] http://nist.gov/speech/tests/tdt/tdt98/

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

[3] http://www.cis.upenn.edu/ dbikel/software.html

[4] http://www.cs.utexas.edu/users/dml/Software/cocluster.html

		All words	LSA	GLSA$_{local}$	GLSA	onlyN	GLSA$_N$	Hybrid
5-10	acc	0.56(0.11)	0.69(0.07)	0.78(0.05)	0.60(0.05)	0.63(0.05)	0.76(0.05)	0.82(0.05)
	F1	0.60(0.09)	0.73(0.05)	0.81(0.04)	0.64(0.05)	0.67(0.05)	0.80(0.04)	0.85(0.04)
50-150	acc	0.75(0.05)	0.73(0.06)	0.80(0.05)	0.70(0.04)	0.68(0.04)	0.80(0.04)	0.87(0.04)
	F1	0.80(0.04)	0.78(0.05)	0.84(0.04)	0.75(0.04)	0.75(0.03)	0.84(0.04)	0.90(0.03)
500-1000	acc	0.95(0.03)	0.98(0.00)	0.99(0.00)	0.97(0.00)	0.97(0.00)	0.99(0.00)	1.00(0.00)
	F1	0.95(0.03)	0.98(0.00)	0.99(0.00)	0.97(0.00)	0.97(0.00)	0.99(0.00)	1.00(0.00)
1000-5000	acc	0.86(0.11)	0.88(0.04)	0.88(0.13)	0.92(0.08)	0.82(0.06)	0.92(0.00)	0.96(0.07)
	F1	0.88(0.07)	0.88(0.03)	0.90(0.09)	0.93(0.06)	0.82(0.04)	0.92(0.00)	0.97(0.05)
5-10$_s$	acc	0.932	0.919	0.986	0.932	0.980	0.980	0.992
	F1	0.933	0.927	0.986	0.932	0.979	0.979	0.992

Table 3: Clustering accuracy (first row) and F1 score (second row) for each indexing scheme. The measures are averaged over 30 random initiations of the clustering algorithm, the standard deviation is shown in brackets. For the last experiment, 5-10$_s$, we used one document per cluster as the initial assignment.

$GLSA_{local}$ on the heterogeneous $5-10$ and $50-150$ subsets and performs similarly for the other two. It supports our intuition that the document's style and word distribution within the collection are important and may get lost, especially if we use a document collection with a different word distribution to estimate the similarities matrix S.

The *tf-idf* indexing with nouns only, $onlyN$, has good performance compared to the all-words indexing. The semantic similarity between nouns seems to be collection independent. The $GLSA_N$ indexing is significantly better than $onlyN$ and *tf-idf* in most cases and performs similar to $GLSA_{local}$. By using $GLSA_N$ we computed the embedding for more nouns that we could keep in the $GLSA_{local}$ and $GLSA$ representations. Nouns convey important topic membership information and it is advantageous to use as many of them as possible.

We observed the same performance relation when we used labels to make the initial cluster assignment, see $5-10_s$ in Table 3. *tf-idf*, GLSA and LSA performed similarly, $GLSA_{local}$ and $GLSA_N$ performed better with the hybrid scheme being the best.

The hybrid indexing significantly outperforms *tf-idf*, LSA and GLSA on three subsets. This shows the benefits of using the spectral embedding to discover the semantic relations between nouns and keeping the rest of the document content as *tf-idf* representation to preserve other indicators of its topic membership. By combining two representations the hybrid indexing scheme defines a more complex notion of similarity between documents. For nouns it uses the semantic proximity in the space of latent semantic classes and for other words it uses term-matching.

References

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

Trevor F. Cox and Micheal A. 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.

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*.

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

J. Munkres. 1957. Algorithms for the assignment and transportation problems. *SIAM*, 5(1):32–38.

Gerard Salton and Michael J. McGill. 1983. *Introduction to Modern Information Retrieval*. McGraw-Hill.

Noam Slonim and Naftali Tishby. 2000. Document clustering using word clusters via the information bottleneck method. In *Research and Development in Information Retrieval*, pages 208–215.