Document Representation and Multilevel Measures of Document Similarity

Irina Matveeva
Dept. of Computer Science
University of Chicago
matveeva@cs.uchicago.edu

Abstract

We present our work on combining large-scale statistical approaches with local linguistic analysis and graph-based machine learning techniques to compute a combined measure of semantic similarity between terms and documents for application in information extraction, question answering, and summarisation.

1 Introduction

Document indexing and representation of term-document relations are crucial for document classification, clustering and retrieval. In the traditional bag-of-words vector space representation of documents (Salton and McGill, 1983) words represent orthogonal dimensions which makes an unrealistic assumption about their independence.

Since document vectors are constructed in a very high dimensional vocabulary space, there has been a considerable interest in low-dimensional document representations to overcome the drawbacks of the bag-of-words document vectors. Latent Semantic Analysis (LSA) (Deerwester et al., 1990) is one of the best known dimensionality reduction algorithms in information retrieval.

In my research, I consider different notions of similarity measure between documents. I use dimensionality reduction and statistical co-occurrence information to define representations that support them.

2 Dimensionality Reduction for Document and Term Representation

A vector space representation of documents is very convenient because it puts documents in a Euclidean space where similarity measures such as inner product and cosine similarity or distance are immediately available. However, these measures will not be effective if they do not have a natural interpretation for the original text data.

I have considered several approaches to computing a vector space representation of text data for which inner product and distance make sense. The general framework is to construct a matrix of pairwise similarities between terms or documents and use appropriate methods of dimensionality reduction to compute low dimensional vectors. The inner product between the resulting vectors must preserve the similarities in the input matrix. The similarities matrix can be computed using different notions of similarity in the input space. Different dimensionality reduction techniques impose different conditions on how the similarities are preserved.

I investigated how external query-based similarity information can be used to compute low dimensional document vectors. Similar to LSA, this approach used weighted bag-of-words document vectors as input which limited its effectiveness. The next step was to develop the Generalized Latent Semantic Analysis framework that allows to compute semantically motivated term and document vectors.

Proceedings of the Human Language Technology Conference of the North American Chapter of the ACL, pages 235–238,
New York, June 2006.

2.1 Document Representation with the Locality Preserving Projection Algorithm

The Locality Preserving Projection algorithm (LPP) (He and Niyogi, 2003) is a graph-based dimensionality reduction algorithm that computes low dimensional document vectors by preserving local similarities between the documents. It requires a vector space representation of documents as input. In addition, it uses the adjacency matrix of the nearest neighbors graph of the data. It can be shown, see (He and Niyogi, 2003), that the Euclidean distance in the LPP space corresponds to similarity in the document space.

The information about the similarity of the input documents is contained in the adjacency matrix of the nearest neighbors graph. In this graph, nodes represent documents and are connected by an edge if the documents are similar. This graph can be constructed using *any* similarity measure between the documents, for example, the query-based similarity between the documents obtained from relevance feedback. The base case is to use inner products between the input document vectors and to connect k nearest neighbors.

We considered several ways of modifying the graph, see (Matveeva, 2004). We used relevance feedback and pseudo relevance feedback from the base line term matching retrieval to identify the top N documents most related to the query. We added edges to the document neighborhood graph to connect these N documents. Our experiments showed that incorporating this external relevance information into the LPP graph improves the performance on the information retrieval tasks, in particular at high levels of recall. Without the use of external information, the performance of the LPP algorithm was comparable to the performance of the LSA algorithm up to recall of 0.6–0.7. At higher levels of recall, LSA achieves a precision that is about 0.1 better than LPP. The precision at high levels of recall seemed to be a weak point of LPP. Fortunately, using the relevance feedback helped to improve the performance in particular in this range of recall.

We found the LPP algorithm to be very sensitive to the graph structure. It confirmed the intuition that the Euclidean distance between the document vectors in the bag-of-words representation is not a good similarity measure. When we added query relevance information to the graph, we introduced a similarity metric on the document space that was closer to the true similarity. However, this information was only partial, because only a subset of the edges reflected this true similarity. The next step was therefore to develop a vector space representation for documents which did not require the bag-of-words representation as input.

2.2 Generalized Latent Semantic Analysis

We developed the Generalized Latent Semantic Analysis (GLSA) framework to compute semantically motivated term and document vectors (Matveeva et al., 2005). We begin with semantically motivated pair-wise term similarities and use dimensionality reduction to compute a vector space representation for terms. Our approach is to focus on similarity between vocabulary terms. We compute representations and similarities for terms and consider documents to be linear combinations of terms. This shift from dual document-term representation to terms has the following motivation.

- Terms offer a much greater flexibility in exploring similarity relations than documents. The availability of large document collections such as the Web offers a great resource for statistical approaches. Recently, co-occurrence based measures of semantic similarity between terms has been shown to improve performance on such tasks as the synonymy test, taxonomy induction, etc. (Turney, 2001; Terra and Clarke, 2003; Chklovski and Pantel, 2004). On the other hand, many semi-supervised and transductive methods based on document vectors cannot yet handle such large document collections.

- 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 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.

2.2.1 GLSA Algorithm

The GLSA algorithm takes as input a document collection C with vocabulary V and a large corpus W. It has the following outline:

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

In step 2 of the GLSA algorithm we used pointwise mutual information (PMI) as the co-occurrence based measure of semantic associations between pairs of the vocabulary terms. We used the singular value decomposition in step 3 to compute GLSA term vectors.

2.2.2 Experimental Evaluation

We used the TOEFL, TS1 and TS2 synonymy tests to demonstrate that the GLSA vector space representation for terms captures their semantic relations, see (Matveeva et al., 2005) for details. Our results demonstrate that similarities between GLSA term vectors achieve better results than PMI scores and outperform the related PMI-IR approach (Turney, 2001; Terra and Clarke, 2003). On the TOEFL test 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 compared to 0.75 in (Terra and Clarke, 2003)).

We also conducted document classification experiments to demonstrate the advantage of the GLSA document vectors (Matveeva et al., 2005). We used a k-nearest neighbors classifier for a set of 5300 documents from 6 dissimilar groups from the 20 news groups data set. The k-nn classifier achieved higher accuracy with the GLSA document vectors than with the traditional tf-idf document vectors, especially with fewer training examples. With 100 training examples, the k-nn classifier with GLSA had 0.75 accuracy vs. 0.58 with the tf-idf document vectors. With 1000 training examples the numbers were 0.81 vs. 0.75.

The inner product between the GLSA document vectors can be used as input to other algorithms. The language modelling approach (Berger and Lafferty, 1999) proved very effective for the information retrieval task. Berger et. al (Berger and Lafferty, 1999) used translation probabilities between the document and query terms to account for synonymy and polysemy. We proposed to use low dimensional term vectors for inducing the translation probabilities between terms (Matveeva and Levow, 2006). We used the same k-nn classification task as above. With 100 training examples, the k-nn accuracy based on tf-idf document vectors was 0.58 and with the similarity based on the language modelling with GLSA term translation probabilities the accuracy was 0.69. With larger training sets the difference in performance was less significant. 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.

3 Work in Progress

Many recent applications such as document summarization, information extraction and question answering require a detailed analysis of semantic relations between terms within and across documents and sentences. Often one has a number of sentences or paragraphs and has to choose the candidate with the highest level of relevance for the topic or question. An additional requirement may be that the information content of the next candidate is different from the sentences that are already chosen.

In these cases, it seems natural to have different levels of document similarity. Two sentences or paragraphs can be similar because they contain information about the same people or events. In this case, the similarity can be based on the number of the named entities they have in common. On the other hand, they can be similar because they contain synonyms or semantically related terms.

I am currently working on a combination of similarity measures between terms to model document similarity. I divide the vocabulary into general vocabulary terms and named entities and compute a separate similarity score for each group of terms. The overall document similarity score is a function of these two scores. To keep the vocabulary size manageable and denoise the data, we only use the content bearing words from the set of the general vocabulary terms. We use a parser to identify nouns and adjectives that participate in three types of syntactic relations: subject, direct object, the head of the noun phrase with an adjective or noun as a modifier for nouns and the modifier of a noun for adjectives. Currently we include only such nouns and adjectives in the set of the content bearing vocabulary terms.

We used the TDT2 collection for preliminary classification experiments. We used a k-nn classifier to classify documents from the 10 most frequent topics. We used tf-idf document vectors indexed with 55,729 general vocabulary words as our baseline. The set of the content bearing words was much smaller and had 13,818 nouns and adjectives. The GLSA document vectors improved the classification accuracy over the baseline and outperformed LSA document vectors. This validates our approach to selecting the content bearing terms and shows the advantage of using the GLSA framework. We are going to extend the set of content bearing words and to include verbs. We will take advantage of the flexibility provided by our framework and use syntax based measure of similarity in the computation of the verb vectors, following (Lin, 1998).

Currently we are using string matching to compute the named entity based measure of similarity. We are planning to integrate more sophisticated techniques in our framework.

4 Conclusion

We developed the GLSA framework for computing semantically motivated term and document vectors. This framework takes advantage of the availability of large document collections and recent research of corpus-based term similarity measures and combines them with dimensionality reduction algorithms.

Different measures of similarity may be required for different groups of terms such as content bearing vocabulary words and named entities. To extend the GLSA approach to computing the document vectors, we use a combination of similarity measures between terms to model the document similarity. This approach defines a fine-grained similarity measure between documents and sentences. Our goal is to develop a multilevel measure of document similarity that will be helpful for summarization and information extraction.

References

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

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

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 and Partha Niyogi. 2003. Locality preserving projections. In *Proc. of NIPS*.

Dekang Lin. 1998. Automatic retrieval and clustering of similar words. In *COLING-ACL*, pages 768–774.

Irina Matveeva and Gina-Anne Levow. 2006. Computing term translation probabilities with generalized latent semantic analysis. In *Proc. of EACL*.

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

Irina Matveeva. 2004. Text representation with the locality preserving projection algorithm for information retrieval task. In *Master's Thesis*.

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

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.