Computing Term Translation Probabilities with Generalized Latent Semantic Analysis

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

Term translation probabilities proved an effective method of semantic smoothing in the language modelling approach to information retrieval tasks. In this paper, we use Generalized Latent Semantic Analysis to compute semantically motivated term and document vectors. The normalized cosine similarity between the term vectors is used as term translation probability in the language modelling framework. Our experiments demonstrate that GLSA-based term translation probabilities capture semantic relations between terms and improve performance on document classification.

1 Introduction

Many recent applications such as document summarization, passage retrieval and question answering require a detailed analysis of semantic relations between terms since often there is no large context that could disambiguate words's meaning.

Many approaches model the semantic similarity between documents using the relations between semantic classes of words, such as representing dimensions of the document vectors 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). They improve the performance on average, but also introduce some instability and thus increased variance (Levow et al., 2005).

The language modelling approach (Ponte and Croft, 1998; Berger and Lafferty, 1999) proved very effective for the information retrieval task. Berger et. al (Berger and Lafferty, 1999) used translation probabilities between terms to account for synonymy and polysemy. However, their model of such probabilities was computationally demanding.

Latent Semantic Analysis (LSA) (Deerwester et al., 1990) is one of the best known dimensionality reduction algorithms. Using a bag-of-words document vectors (Salton and McGill, 1983), it computes a dual representation for terms and documents in a lower dimensional space. The resulting document vectors reside in the space of latent semantic concepts which can be expressed using different words. The statistical analysis of the semantic relatedness between terms is performed implicitly, in the course of a matrix decomposition.

In this project, we propose to use a combination of dimensionality reduction and language modelling to compute the similarity between documents. We compute term vectors using the Generalized Latent Semantic Analysis (Matveeva et al., 2005). This method uses co-occurrence based measures of semantic similarity between terms to compute low dimensional term vectors in the space of latent semantic concepts. The normalized cosine similarity between the term vectors is used as term translation probability.

2 Term Translation Probabilities in Language Modelling

The language modelling approach (Ponte and Croft, 1998) 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 $\mathbf{q} = (q_1, ..., q_m)$, the likelihood of the query is estimated using the document's distribution: $p(\mathbf{q}|d) = \prod_1^m p(q_i|d)$, where

q_i are query terms. Relevant documents maximize $p(d|\mathbf{q}) \propto p(\mathbf{q}|d)p(d)$.

Many relevant documents may not contain the same terms as the query. However, they may contain terms that are semantically related to the query terms and thus have high probability of being "translations", i.e. re-formulations for the query words.

Berger et. al (Berger and Lafferty, 1999) introduced translation probabilities between words into the document-to-query model as a way of semantic smoothing of the conditional word probabilities. Thus, they query-document similarity is computed as

$$p(\mathbf{q}|d) = \prod_{i}^{m} \sum_{w \in d} t(q_i|w)p(w|d). \qquad (1)$$

Each document word w is a translation of a query term q_i with probability $t(q_i|w)$. 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 use the Generalized Latent Semantic Analysis to compute the translation probabilities.

2.1 Document Similarity

We propose to use low dimensional term vectors for inducing the translation probabilities between terms. We postpone the discussion of how the term vectors are computed to section 2.2. To evaluate the validity of this approach, we applied it to document classification.

We used two methods of computing the similarity between documents. First, we computed the language modelling score using term translation probabilities. Once the term vectors are computed, the document vectors are generated as linear combinations of term vectors. Therefore, we also used the cosine similarity between the documents to perform classificaiton.

We computed the language modelling score of a test document d relative to a training document d_i as

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

Appropriately normalized values of the cosine similarity measure between pairs of term vectors $\cos(\vec{v}, \vec{w})$ are used as the translation probability between the corresponding terms $t(v|w)$.

In addition, we used the cosine similarity between the document vectors

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

where $\alpha_w^{d_i}$ and $\beta_v^{d_j}$ represent the weight of the terms w and v with respect to the documents d_i and d_j, respectively.

In this case, the inner products between the term vectors are also used to compute the similarity between the document vectors. Therefore, the cosine similarity between the document vectors also depends on the relatedness between pairs of terms.

We compare these two document similarity scores to the cosine similarity between bag-of-word document vectors. Our experiments show that these two methods offer an advantage for document classification.

2.2 Generalized Latent Semantic Analysis

We use the Generalized Latent Semantic Analysis (GLSA) (Matveeva et al., 2005) to compute semantically motivated term vectors.

The GLSA algorithm computes the term vectors for the vocabulary of the document collection C with vocabulary V using 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, $U^T \in R^{k \times |V|}$

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.

In step 2 we used point-wise mutual information (PMI) as the co-occurrence based measure of semantic associations between pairs of the vocabulary terms. PMI has been successfully applied to semantic proximity tests for words (Turney, 2001; Terra and Clarke, 2003) and was also successfully used as a measure of term similarity to compute document clusters (Pantel and Lin, 2002). In our preliminary experiments, the GLSA with PMI showed a better performance than with other co-occurrence based measures such as the likelihood ratio, and χ^2 test.

PMI between random variables representing two words, w_1 and w_2, is computed as

$$PMI(w_1, w_2) = \log \frac{P(W_1 = 1, W_2 = 1)}{P(W_1 = 1)P(W_2 = 1)}. \quad (4)$$

We used the singular value decomposition (SVD) in step 3 to compute GLSA term vectors.

LSA (Deerwester et al., 1990) and some other related dimensionality reduction techniques, e.g. Locality Preserving Projections (He and Niyogi, 2003) compute a dual document-term representation. The main advantage of GLSA is that it focuses on term vectors which allows for a greater flexibility in the choice of the similarity matrix.

3 Experiments

The goal of the experiments was to understand whether the GLSA term vectors can be used to model the term translation probabilities. We used a simple k-NN classifier and a basic baseline to evalute the performance. We used the GLSA-based term translation probabilities within the language modelling framework and GLSA document vectors.

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. Here we used 2 sets of 6 news groups. $Group_d$ contained documents from dissimilar news groups[1], with a total of 5300 documents. $Group_s$ contained documents from more similar news groups[2] and had 4578 documents.

3.1 GLSA Computation

To collect the co-occurrence statistics for the similarities matrix S we used the English Gigaword collection (LDC). We used 1,119,364 New York Times articles labeled "story" with 771,451 terms. We used the Lemur toolkit[3] to tokenize and index the document; we used stemming and a list of stop words. Unless stated otherwise, for the GLSA methods we report the best performance over different numbers of embedding dimensions.

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 (Terra and Clarke, 2003). We used the window of size 4.

	$Group_d$			$Group_s$		
#L	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.

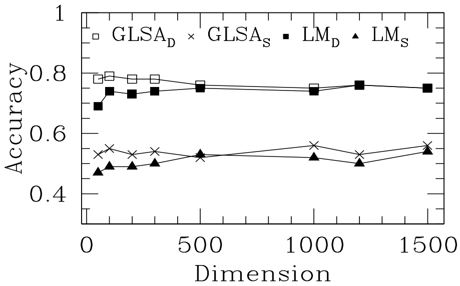

Figure 1: k-NN with 400 training documents.

3.2 Classification Experiments

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. Other weighting schemes such as maximum likelihood and Laplace smoothing did not improve results.

Table 1 shows the results. We computed the score between the training and test documents using two approaches: cosine similarity between the GLSA 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

[1] os.ms, sports.baseball, rec.autos, sci.space, misc.forsale, religion-christian

[2] politics.misc, politics.mideast, politics.guns, religion.misc, religion.christian, atheism

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

LM). We used the term frequency as an estimate for $p(w|d)$. To compute the matrix of translation probabilities P, where $P[i][j] = t(t_j|t_i)$ for the $\mathrm{LM}_{\mathrm{GLSA}}$ approach, we first obtained the matrix $\hat{P}[i][j] = \cos(\vec{t_i}, \vec{t_j})$. We set the negative and zero entries in $\hat{P}$ 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 usually peaked at around 300-500 dimensions which is in line with results for other SVD-based approaches (Deerwester et al., 1990). When the highest accuracy was achieved at higher dimensions, the increase after 500 dimensions was rather small, as illustrated in Figure 1.

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.

4 Conclusion and Future Work

We used the GLSA to compute term translation probabilities as a measure of semantic similarity between documents. We showed that the GLSA term-based document representation and GLSA-based term translation probabilities improve performance on document classification.

The GLSA term vectors were computed for all vocabulary terms. However, different measures of similarity may be required for different groups of terms such as content bearing general vocabulary words and proper names as well as other named entities. Furthermore, different measures of similarity work best for nouns and verbs. To extend this approach, we will use a combination of similarity measures between terms to model the document similarity. We will divide the vocabulary into general vocabulary terms and named entities and compute a separate similarity score for each of the group of terms. The overall similarity score is a function of these two scores. In addition, we will use the GLSA-based score together with syntactic similarity to compute the similarity between the general vocabulary terms.

References

Ron Bekkerman, Ran El-Yaniv, and Naftali Tishby. 2003. Distributional word clusters vs. words for text categorization.

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

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*.

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.

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*.

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.

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.