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. We use Generalized Latent Semantic Analysis to compute semantically motivated term and document vectors. Normalized cosine similarity between term vectors is used as term translation probability. 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. The language modelling approach (Ponte and Croft, 1998) uses translation probabilities between terms to account for synonymy and polysemy.

In the language modelling approach documents define a multinomial probability distribution $p(w|d)$ over the vocabulary. The conditional 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 $p(d|\mathbf{q}) \propto p(\mathbf{q}|d)p(d)$. Berger et. al (Berger and Lafferty, 1999) introduced translation probabilities between words as a way of semantic smoothing of the conditional word probabilities. The estimation of the translation probabilities is, however, a difficult task.

We use the Generalized Latent Semantic Analysis (GLSA) (Matveeva et al., 2005) to induce translation probabilities. GLSA is a dimensionality reduction framework that computes term vectors in the latent semantic space so that the cosine similarities preserve the pair-wise term similarities in the input space. We use appropriately normalized cosine similarities between GLSA term vectors as term translation probabilities.

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.

2 Experiments

We used a k-NN classification experiment to validate our approach. We used the 20 newsgroups data set. We computed GLSA term vectors for 9732 terms that occurred in at least 15 documents and used them for document representation. Here we used 2 sets of 6 news groups. $Group_d$ contained documents from dissimilar news groups[1], with 5300 documents. $Group_s$ contained documents from more similar news groups[2] and had 4578 docu-

[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

	$Group_d$			$Group_s$		
#L	tf-idf	Glsa	LM	tf-idf	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.

ments. We used the Lemur toolkit[3] to tokenize and index the documents; we used stemming and a list of stop words. For the GLSA methods we report the best performance over different numbers of embedding dimensions. 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 the cosine similarity between the bag-of-words document vectors weighted with tf-idf.

We computed the score between a training document d_i and a test document d_j using the language modelling score which included the translation probabilities between the terms, as in Equation 1, and cosine similarity between the GLSA document vectors, as in Equation 2. We used term frequency as an estimate for $p(w|d)$.

$$p(d_j|d_i) = \prod_{v \in d_j} \sum_{w \in d_i} t(v|w) p(w|d_i), \qquad (1)$$

where $t(v|w) \propto \langle \vec{v}, \vec{w} \rangle$. $\vec{v}$ and $\vec{w}$ are GLSA term vectors for terms v and w.

$$\langle \vec{d_j}, \vec{d_i} \rangle = \sum_{v \in d_j} \sum_{w \in d_i} \alpha_v^{d_j} \beta_w^{d_i} \langle \vec{v}, \vec{w} \rangle, \qquad (2)$$

where $\alpha_v^{d_j}$ and $\beta_w^{d_i}$ represent the weight of the terms v and w with respect to the documents d_j and d_i.

2.1 Results

Table 1 shows the classification accuracy using the cosine between the tf-idf document vectors (tf-idf), the cosine between the GLSA document vectors (GLSA) and the language modelling score with the GLSA based translation probabilities (LM) for different sizes of the training set ($\#L$). For both groups of documents, GLSA and LM outperform the *tf-idf* document vectors. As expected, the classification task was more difficult for the similar news groups. In both cases, the advantage is more significant with smaller sizes of the training set. There is no significant difference in the performance of GLSA and LM for the similar newsgroups. GLSA had a higher accuracy with smaller sizes of the training sets for the dissimilar newsgroups. We are planning larger classification experiment to investigate the difference between these two approaches.

3 Conclusion and Future Work

We proposed a new method of computing term translation probabilities in the language modelling framework. We have shown 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

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

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

Jay M. Ponte and W. Bruce Croft. 1998. A language modeling approach to information retrieval. In *Proc. of the 21st ACM SIGIR*.

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