Query Segmentation Based on Eigenspace Similarity

Chao Zhang [†] [‡] **Nan Sun** [‡] **Xia Hu** [‡] **Tingzhu Huang** [†] **Tat-Seng Chua** [‡]

[†]School of Applied Math
University of Electronic Science
and Technology of China,
Chengdu, 610054, P.R. China
zhangcha@comp.nus.edu.sg
tzhuang@uestc.edu.cn

[‡]School of Computing
National University of Singapore,
Computing 1, Singapore 117590
{sunn,huxia,chuats}@comp.nus.edu.sg

Abstract

Query segmentation is essential to query processing. It aims to tokenize query words into several semantic segments and help the search engine to improve the precision of retrieval. In this paper, we present a novel unsupervised learning approach to query segmentation based on principal eigenspace similarity of query-word-frequency matrix derived from web statistics. Experimental results show that our approach could achieve superior performance of 35.8% and 17.7% in F-measure over the two baselines respectively, i.e. MI (Mutual Information) approach and EM optimization approach.

1 Introduction

People submit concise word-sequences to search engines in order to obtain satisfying feedback. However, the word sequences are generally ambiguous and often fail to convey the exact information to search engine, thus severely, affecting the performance of the system. For example, given the query "free software testing tools download". A simple bag-of-words query model cannot analyze "software testing tools" accurately. Instead, it returns "free software" or "free download" which are high frequency web phrases. Therefore, how to segment a query into meaningful semantic components for implicit description of user's intention is an important issue both in natural language processing and information retrieval fields.

There are few related studies on query segmentation in spite of its importance and applicability in many query analysis tasks such as query suggestion, query substitution, etc. To our knowledge, three approaches have been studied in previous works: MI (Mutual Information) approach (Jones et al., 2006; Risvik et al., 2003), supervised learning approach (Bergsma and Wang, 2007) and EM optimization approach (Tan and Peng, 2008). However, MI approach calculates MI value just between two adjacent words that cannot handle long entities. Supervised learning approach requires a sufficiently large number of labeled training data, which is not conducive in real applications. EM algorithm often converges to a local maximum that depends on the initial conditions. There are also many relevant research on Chinese word segmentation (Teahan et al., 2000; Peng and Schuurmans, 2001; Xu et al., 2008). However, they cannot be applied directly to query segmentation (Tan and Peng, 2008).

Under this scenario, we propose a novel unsupervised approach for query segmentation. Differing from previous work, we first adopt the n-gram model to estimate the query term's frequency matrix based on word occurrence statistics on the web. We then devise a new strategy to select principal eigenvectors of the matrix. Finally we calculate the similarity of query words for segmentation. Experimental results demonstrate the effectiveness of our approach as compared to two baselines.

2 Methodology

In this Section, we introduce our proposed query segmentation approach, which is based on query word frequency matrix principal eigenspace similarity. To facilitate understanding, we first present a general overview of our approach in Section 2.1 and then describe the details in Section 2.2-2.5.

2.1 Overview

Figure 1 briefly shows the main procedure of our proposed query segmentation approach. It starts with a query which consists of a vector of words$\{w_1 w_2 \cdots w_n\}$. Our approach first build a query-word frequency matrix M based on web statistics to describe the relationship between any

Proceedings of the ACL-IJCNLP 2009 Conference Short Papers, pages 185–188,
Suntec, Singapore, 4 August 2009.

two query words (Step 1). After decomposing M (step 2), the parameter k which defines the number of segments in the query is estimate in Step 3. Besides, a principal eigenspace of M is built and the projection vectors($\{\alpha_i\}, i \in [1, n]$) associated with each query-word are obtained (Step 4). Similarities between projection vectors are then calculated, which determine whether the corresponding two words should be segmented together (Step5). If the number of segmented components is not equal to k, our approach modifies the threshold δ and repeats steps 5 and 6 until the correct k number of segmentations are obtained(Step 7).

Input:	one n words query: $w_1 w_2 \cdots w_n$;
Output:	k segmented components of query;
Step 1:	Build a frequency matrix M (Section 2.2);
Step 2:	Decompose M into sorted eigenvalues and eigenvectors;
Step 3:	Estimate parameter k (Section 2.4);
Step 4:	Build principal eigenspace with first k eigenvectors and get the projection ($\{\alpha_i\}$) of M in principal eigenspace (Section 2.3);
Step 5:	Segment the query: if $(\alpha_i \cdot \alpha_j^T)/(\|\alpha_i\| \cdot \|\alpha_j\|) \geq \delta$, segment w_i and w_j together (Section 2.5)
Step 6:	If the number of segmented parts does not equal to k, modify δ, go to step 5;
Step 7:	output the right segmentations

Figure 1: Query Segmentation based on query-word-frequency matrix eigenspace similarity

2.2 Frequency Matrix

Let $W = w_1, w_2, \cdots, w_n$ be a query of n words. We can build the relationships of any two words using a symmetric matrix: $M = \{m_{i,j}\}_{n \times n}$

$$m_{i,j} = \begin{cases} F(w_i) & \text{if } i = j \\ F(w_i w_{i+1} \cdots w_j) & \text{if } i < j \\ m_{j,i} & \text{if } i > j \end{cases} \quad (1)$$

$$F(w_i w_{i+1} \cdots w_j) = \frac{count(w_i w_{i+1} \cdots w_j)}{\sum_{i=1}^{n} w_i} \quad (2)$$

Here $m_{i,j}$ denotes the correlation between $(w_i \cdots w_{j-1})$ and w_j, where $(w_i \cdots w_{j-1})$ means a sequence and w_j is a word. Considering the difference of each matrix element $m_{i,j}$, we normalize $m_{i,j}$ with:

$$m_{i,j} = 2 \cdot m_{i,j}/(m_{i,i} + m_{j,j}) \quad (3)$$

$F(\cdot)$ is a function measuring the frequency of query words or sequences. To improve the precision of measurement and reduce the computation cost, we adopt the approach proposed by (Wang et al., 2007) here. First, we extract the relevant documents associated with the query via Google Soap Search API. Second, we count the number of all possible n-gram sequences which are highlighted in the titles and snippets of the returned documents. Finally, we use Eqn.(2) to estimate the value of $m_{i,j}$.

2.3 Principal Eigenspace

Although matrix M depicts the correlation of query words, it is rough and noisy. Under this scenario, we transform M into its principal eigenspace which is spanned by k largest eigenvectors, and each query word is denoted by the corresponding eigenvector in the principal eigenspace.

Since M is a symmetric positive definite matrix, its eigenvalues are real numbers and the corresponding eigenvectors are non-zero and orthotropic to each other. Here, we denote the eigenvalues of M as : $\lambda(M) = \{\lambda_1, \lambda_2, \cdots, \lambda_n\}$ and $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$. All eigenvalues of M have corresponding eigenvectors:$V(M) = \{x_1, x_2, \cdots, x_n\}$.

Suppose that principal eigenspace $\overline{M}(\overline{M} \in \mathcal{R}^{n \times k})$ is spanned by the first k eigenvectors, i.e. $\overline{M} = Span\{x_1, x_2, \cdots x_k\}$, then row i of M can be represented by vector α_i which denotes the i-th word for similarity calculation in Section 2.5, and α_i is derived from:

$$\{\alpha_1^T, \alpha_2^T, \cdots, \alpha_n^T\}^T = \{x_1, x_2, \cdots, x_k\} \quad (4)$$

Section 2.4 discusses the details of how to select the parameter k.

2.4 Parameter k Selection

PCA (principal component analysis) (Jolliffe, 2002) often selects k principal components by the following criterion:

k is the smallest integer which satisfies:

$$\frac{\sum_{i=1}^{k} \lambda_i}{\sum_{i=1}^{n} \lambda_i} \geq Threshold \quad (5)$$

where n is the number of eigenvalues. When $\lambda_k \gg \lambda_{k+1}$, Eqn.(5) is very effective. However, according to the Gerschgorin circle theorem, the non-diagonal values of M are so small that the eigenvalues cannot be distinguished easily. Under this circumstance, a prefixed threshold is too restrictive to be applied in complex situations. Therefore a function of n is introduced into the threshold as follows:

$$\frac{\sum_{i=1}^{k} \lambda_i}{\sum_{i=1}^{n} \lambda_i} \geq (\frac{n-1}{n})^2 \quad (6)$$

If k eigenvalues are qualified to be the principal components, then the threshold in Eqn.(5) cannot be lower than 0.5, and need not be higher than $\frac{n-1}{n}$. If the length of the shortest query we segmented is 4, we choose $(\frac{n-1}{n})^2$ because it will be smaller than $\frac{n-1}{n}$ and larger than 0.5 with n no smaller than 4.

The k eigenvectors will be used to segment the query into k meaningful segments (Weiss, 1999; Ng et al., 2001). In the k-dimensional principal eigenspace, each dimension of the space describes a semantic concept of the query. When one eigenvalue is bigger, the corresponding dimension contains more query words.

2.5 Similarity Computation

If the word i and word j are co-occurrence, α_i and α_j are approximately parallel in the principal eigenspace; otherwise, they are approximately orthogonal to each other. Hence, we measure the similarity of α_i and α_j with inner-product to perform the segmentation (Weiss, 1999; Ng et al., 2001). Selecting a proper threshold δ, we segment the query using Eqn.(7):

$$S(w_i, w_j) = \begin{cases} 1, & (\alpha_i \cdot \alpha_j^T)/(\|\alpha_i\| \cdot \|\alpha_j\|) \geq \delta \\ 0, & (\alpha_i \cdot \alpha_j^T)/(\|\alpha_i\| \cdot \|\alpha_j\|) < \delta \end{cases} \quad (7)$$

If $S(w_i, w_j) = 1$, w_i and w_j should be segmented together, otherwise, w_i and w_j belong to different semantic concepts respectively. Here, we denote the total number of segments of the query as integer m.

As mentioned in Section 2.4, m should be equal to k, therefore, the threshold δ is modified by k and m. We set the initial value $\delta = 0.5$ and modify it with binary search method until $m = k$. If k is larger than m, it means δ is too small to be a proper threshold, i.e. some segments should be further segmented. Otherwise, δ is too large that it should be reduced.

3 Experiments

3.1 Data set

We experiment on the data set published by (Bergsma and Wang, 2007). This data set comprises 500 queries which were randomly taken from the AOL search query database and each query. These queries are all segmented manually by three annotators (the results are referred as **A**, **B** and **C**).

We evaluate our results on the five test data sets (Tan and Peng, 2008), i.e. we use A, B, C, the intersection of three annotator's results (referred to as **D**) and the conjunction of three annotator's results (referred to as **E**). Besides, three evaluation metrics are used in our experiments (Tan and Peng, 2008; Peng and Schuurmans, 2001), i.e. *Precision* (referred to as Prec), *Recall and F-Measure* (referred to as F-mea).

3.2 Experimental results

Two baselines are used in our experiments: one is MI based method (referred to as MI), and the other is EM optimization (referred to as EM). Since the EM proposed in (Tan and Peng, 2008) is implemented with Yahoo! web corpus and only Google Soap Search API is available in our study, we adopt $t\text{-}test$ to evaluate the performance of MI with Google data (referred to as MI(G)) and Yahoo! web corpus (referred to as MI(Y)). With the values of MI(Y) and MI(G) in Table 1 we get the $p\text{-}value$ ($p = 0.316 \gg 0.05$), which indicates that the performance of MI with different corpuses has no significant difference. Therefore, we can deduce that, the two corpuses have little influence on the performance of the approaches. Here, we denote our approach as "ES", i.e. Eigenspace Similarity approach.

Table 1 presents the performance of the three approaches, i.e. MI (MI(Y) and MI(G)), EM and our proposed ES on the five test data sets using the three mentioned metrics. From Table 1 we find that ES achieves significant improvements as compared to the other two methods in any metric and data set we used.

For further analysis, we compute statistical performance on mathematical expectation and standard deviation as shown in Figure 2. We observe a consistent trend of the three metrics increasing from left to right as shown in Figure 2, i.e. EM performs better than MI and ES is the best among the three approaches.

		MI(Y)	MI(G)	EM	ES
A	Prec	0.469	0.548	0.562	0.652
	Recall	0.534	0.489	0.555	0.699
	F-mea	0.499	0.517	0.558	**0.675**
B	Prec	0.408	0.449	0.568	0.632
	Recall	0.472	0.391	0.578	0.659
	F-mea	0.438	0.418	0.573	**0.645**
C	Prec	0.451	0.503	0.558	0.614
	Recall	0.519	0.440	0.561	0.649
	F-mea	0.483	0.469	0.559	**0.631**
D	Prec	0.510	0.574	0.640	0.772
	Recall	0.550	0.510	0.650	0.826
	F-mea	0.530	0.540	0.645	**0.798**
E	Prec	0.582	0.672	0.715	0.834
	Recall	0.654	0.734	0.721	0.852
	F-mea	0.616	0.702	0.718	**0.843**

Table 1: Performance of different approaches.

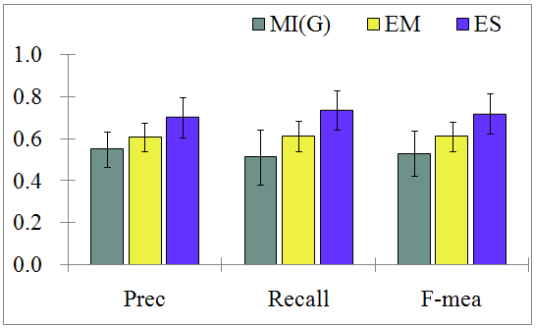

Figure 2: Statistical performance of approaches

First, we observe that, EM (Prec: 0.609, Recall: 0.613, F-mea: 0.611) performs much better than MI (Prec: 0.549, Recall: 0.513, F-mea: 0.529). This is because EM optimizes the frequencies of query words with EM algorithms. In addition, it should be noted that, the recall of MI is especially unsatisfactory, which is caused by its shortcoming on handling long entities.

Second, when compared with EM, ES also has more than 15% increase in the three reference metrics (15.1% on Prec, 20.2% on Recall and 17.7% on F-mea). Here all increases are statistically significant with $p\text{-}value$ closed to 0. In depth analysis indicates that this is because ES makes good use of the frequencies of query words in its principal eigenspace, while EM algorithm trains the observed data (frequencies of query words) by simply maximizing them using maximum likelihood.

4 Conclusion and Future work

We proposed an unsupervised approach for query segmentation. After using n-gram model to estimate term frequency matrix using term occurrence statistics from the web, we explored a new method to select principal eigenvectors and calculate the similarities of query words for segmentation. Experiments demonstrated the effectiveness of our approach, with significant improvement in segmentation accuracy as compared to the previous works.

Our approach will be capable of extracting semantic concepts from queries. Besides, it can extended to Chinese word segmentation. In future, we will further explore a new method of parameter k selection to achieve higher performance.

References

S. Bergsma and Q. I. Wang. 2007. *Learning Noun Phrase Query Segmentation*. In Proc of EMNLP-CoNLL

R. Jones, B. Rey, O. Madani, and W. Greiner. 2006. *Generating query substitutions*. In Proc of WWW.

I.T. Jolliffe. 2002. *Principal Component Analysis*. Springer, NY, USA.

Andrew Y. Ng, Michael I. Jordan, Yair Weiss. 2001. *On spectral clustering: Analysis and an algorithm* In Proc of NIPS.

F. Peng and D. Schuurmans. 2001. *Self-Supervised Chinese Word Segmentation*. Proc of the 4th Int'l Conf. on Advances in Intelligent Data Analysis.

K. M. Risvik, T. Mikolajewski, and P. Boros. 2003. *Query Segmentation for Web Search*. In Proc of WWW.

Bin Tan, Fuchun Peng. 2008. *Unsupervised Query Segmentation Using Generative Language Models and Wikipedia.* In Proc of WWW.

W. J. Teahan Rodger Mcnab Yingying Wen Ian H. Witten . 2000. *A compression-based algorithm for Chinese word segmentation* Computational Linguistics.

Xin-Jing Wang, Wen Liu, Yong Qin. 2007. *A Search-based Chinese Word Segmentation Method.* In Proc of WWW.

Yair Weiss. 1999. *Segmentation using eigenvectors: a unifying view.* Proc. IEEE Int'l Conf. Computer Vision, vol. 2, pp. 975-982.

Jia Xu, Jianfeng Gao, Kristina Toutanova, Hermann. 2008. *Bayesian Semi-Supervised Chinese Word Segmentation for Statistical Machine Translation.* In Proc of COLING.