Latent Semantic Tensor Indexing for Community-based Question Answering

Xipeng Qiu, Le Tian, Xuanjing Huang
Fudan University, 825 Zhangheng Road, Shanghai, China
xpqiu@fudan.edu.cn, tianlefdu@gmail.com, xjhuang@fudan.edu.cn

Abstract

Retrieving similar questions is very important in community-based question answering(CQA). In this paper, we propose a unified question retrieval model based on latent semantic indexing with tensor analysis, which can capture word associations among different parts of CQA triples simultaneously. Thus, our method can reduce lexical chasm of question retrieval with the help of the information of question content and answer parts. The experimental result shows that our method outperforms the traditional methods.

1 Introduction

Community-based (or collaborative) question answering(CQA) such as Yahoo! Answers[1] and Baidu Zhidao[2] has become a popular online service in recent years. Unlike traditional question answering (QA), information seekers can post their questions on a CQA website which are later answered by other users. However, with the increase of the CQA archive, there accumulate massive duplicate questions on CQA websites. One of the primary reasons is that information seekers cannot retrieve answers they need and thus post another new question consequently. Therefore, it becomes more and more important to find semantically similar questions.

The major challenge for CQA retrieval is the lexical gap (or lexical chasm) among the questions (Jeon et al., 2005b; Xue et al., 2008), as shown in Table 1. Since question-answer pairs are usually short, the word mismatching problem is especially important. However, due to the lexical gap between questions and answers as well as spam typically existing in user-generated content, filtering and ranking answers is very challenging.

Query:
Q: Why is my laptop screen blinking?
Expected:
Q1: How to troubleshoot a flashing screen on an LCD monitor?
Not Expected:
Q2: How to blinking text on screen with PowerPoint?

Table 1: An example on question retrieval

The earlier studies mainly focus on generating redundant features, or finding textual clues using machine learning techniques; none of them ever consider questions and their answers as relational data but instead model them as independent information. Moreover, they only consider the answers of the current question, and ignore any previous knowledge that would be helpful to bridge the lexical and se mantic gap.

In recent years, many methods have been proposed to solve the word mismatching problem between user questions and the questions in a QA archive(Blooma and Kurian, 2011), among which the translation-based (Riezler et al., 2007; Xue et al., 2008; Zhou et al., 2011) or syntactic-based approaches (Wang et al., 2009) methods have been proven to improve the performance of CQA retrieval.

However, most of these approaches used

[1] http://answers.yahoo.com/
[2] http://zhidao.baidu.com/

Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, pages 434–439,
Sofia, Bulgaria, August 4-9 2013.

pipeline methods: (1) modeling word association; (2) question retrieval combined with other models, such as vector space model (VSM), Okapi model (Robertson et al., 1994) or language model (LM). The pipeline methods often have many non-trivial experimental setting and result to be very hard to reproduce.

In this paper, we propose a novel unified retrieval model for CQA, **latent semantic tensor indexing** (LSTI), which is an extension of the conventional latent semantic indexing (LSI) (Deerwester et al., 1990). Similar to LSI, LSTI can integrate the two detached parts (modeling word association and question retrieval) into a single model.

In traditional document retrieval, LSI is an effective method to overcome two of the most severe constraints on Boolean keyword queries: synonymy, that is, multiple words with similar meanings, and polysemy, or words with more than one meanings.

Usually in a CQA archive, each entry (or question) is in the following triple form:⟨**question title, question content, answer**⟩. Because the performance based solely on the content or the answer part is less than satisfactory, many works proposed that additional relevant information should be provided to help question retrieval(Xue et al., 2008). For example, if a question title contains the keyword "why", the CQA triple, which contains "because" or "reason" in its answer part, is more likely to be what the user looks for.

Since each triple in CQA has three parts, the natural representation of the CQA collection is a three-dimensional array, or 3rd-order tensor, rather than a matrix. Based on the tensor decomposition, we can model the word association simultaneously in the pairs: question-question, question-body and question-answer.

The rest of the paper is organized as follows: Section 3 introduces the concept of LSI. Section 4 presents our method. Section 5 describes the experimental analysis. Section 6 concludes the paper.

2 Related Works

There are some related works on question retrieval in CQA. Various query expansion techniques have been studied to solve word mismatch problems between queries and documents. The early works on question retrieval can be traced back to finding similar questions in Frequently Asked Questions (FAQ) archives, such as the FAQ finder (Burke et al., 1997), which usually used statistical and semantic similarity measures to rank FAQs.

Jeon et al. (2005a; 2005b) compared four different retrieval methods, i.e., the vector space model(Jijkoun and de Rijke, 2005), the Okapi BM25 model (Robertson et al., 1994), the language model, and the translation model, for question retrieval on CQA data, and the experimental results showed that the translation model outperforms the others. However, they focused only on similarity measures between queries (questions) and question titles.

In subsequent work (Xue et al., 2008), a translation-based language model combining the translation model and the language model for question retrieval was proposed. The results showed that translation models help question retrieval since they could effectively address the word mismatch problem of questions. Additionally, they also explored answers in question retrieval.

Duan et al. (2008) proposed a solution that made use of question structures for retrieval by building a structure tree for questions in a category of Yahoo! Answers, which gave more weight to important phrases in question matching.

Wang et al. (2009) employed a parser to build syntactic trees for questions, and questions were ranked based on the similarity between their syntactic trees and that of the query question.

It is worth noting that our method is totally different to the work (Cai et al., 2006) of the same name. They regard documents as matrices, or the second order tensors to generate a low rank approximations of matrices (Ye, 2005). For example, they convert a $1,000,000$-dimensional vector of word space into a 1000×1000 matrix. However in our model, a document is still represented by a vector. We just project a higher-dimensional vector to a lower-dimensional vector, but not a matrix in Cai's model. A 3rd-order tensor is

also introduced in our model for better representation for CQA corpus.

3 Latent Semantic Indexing

Latent Semantic Indexing (LSI) (Deerwester et al., 1990), also called Latent Semantic Analysis (LSA), is an approach to automatic indexing and information retrieval that attempts to overcome these problems by mapping documents as well as terms to a representation in the so-called latent semantic space.

The key idea of LSI is to map documents (and by symmetry terms) to a low dimensional vector space, the latent semantic space. This mapping is computed by decomposing the term-document matrix N with SVD, $N = U\Sigma V^t$, where U and V are orthogonal matrices $U^tU = V^tV = I$ and the diagonal matrix Σ contains the singular values of N. The LSA approximation of N is computed by just keep the largest K singular values in Σ, which is rank K optimal in the sense of the L^2-norm.

LSI has proven to result in more robust word processing in many applications.

4 Tensor Analysis for CQA

4.1 Tensor Algebra

We first introduce the notation and basic definitions of multilinear algebra. Scalars are denoted by lower case letters $(a, b, \dots)$, vectors by bold lower case letters $(\mathbf{a}, \mathbf{b}, \dots)$, matrices by bold upper-case letters $(\mathbf{A}, \mathbf{B}, \dots)$, and higher-order tensors by calligraphic upper-case letters $(\mathcal{A}, \mathcal{B}, \dots)$.

A tensor, also known as n-way array, is a higher order generalization of a vector (first order tensor) and a matrix (second order tensor). The order of tensor $\mathcal{D} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ is N. An element of $\mathcal{D}$ is denoted as $d_{i_1,\dots,i_N}$.

An Nth-order tensor can be flattened into a matrix by N ways. We denote the matrix $\mathbf{D}(n)$ as the mode-n flattening of $\mathcal{D}$ (Kolda, 2002).

Similar with a matrix, an Nth-order tensor can be decomposed through "N-mode singular value decomposition (SVD)", which is a an extension of SVD that expresses the tensor as the mode-n product of N-orthogonal spaces.

$$\mathcal{D} = \mathcal{Z} \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \cdots \times_n \mathbf{U}_n \cdots \times_N \mathbf{U}_N. \quad (1)$$

Tensor $\mathcal{Z}$, known as the core tensor, is analogous to the diagonal singular value matrix in conventional matrix SVD. $\mathcal{Z}$ is in general a full tensor. The core tensor governs the interaction between the mode matrices $\mathbf{U}_n$, for $n = 1, \dots, N$. Mode matrix $\mathbf{U}_n$ contains the orthogonal left singular vectors of the mode-n flattened matrix $\mathbf{D}(n)$.

The N-mode SVD algorithm for decomposing $\mathcal{D}$ is as follows:

1. For $n = 1, \dots, N$, compute matrix $\mathbf{U}_n$ in Eq.(1) by computing the SVD of the flattened matrix $\mathbf{D}(n)$ and setting $\mathbf{U}_n$ to be the left matrix of the SVD.

2. Solve for the core tensor as follows $\mathcal{Z} = \mathcal{D} \times_1 \mathbf{U}_1^T \times_2 \mathbf{U}_2^T \cdots \times_n \mathbf{U}_n^T \cdots \times_N \mathbf{U}_N^T$.

4.2 CQA Tensor

Given a collection of CQA triples, $\langle q_i, c_i, a_i \rangle$ $(i = 1, \dots, K)$, where q_i is the question and c_i and a_i are the content and answer of q_i respectively. We can use a 3-order tensor $\mathcal{D} \in \mathbb{R}^{K \times 3 \times T}$ to represent the collection, where T is the number of terms. The first dimension corresponds to entries, the second dimension, to parts and the third dimension, to the terms.

For example, the flattened matrix of CQA tensor with "terms" direction is composed by three sub-matrices $\mathbf{M_{Title}}$, $\mathbf{M_{Content}}$ and $\mathbf{M_{Answer}}$, as was illustrated in Figure 1. Each sub-matrix is equivalent to the traditional document-term matrix.

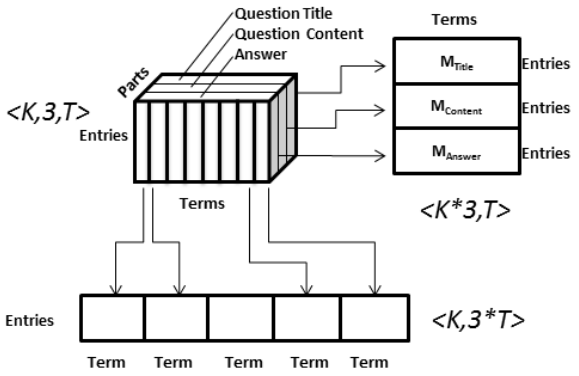

Figure 1: Flattening CQA tensor with "terms" (right matrix)and "entries" (bottom matrix)

Denote $p_{i,j}$ to be part j of entry i. Then we

have the term frequency, defined as follows.

$$\mathrm{tf_{i,j,k}} = \frac{n_{i,j,k}}{\sum_i n_{i,j,k}}, \tag{2}$$

where $n_{i,j,k}$ is the number of occurrences of the considered term (t_k) in $p_{i,j}$, and the denominator is the sum of number of occurrences of all terms in $p_{i,j}$.

The inverse document frequency is a measure of the general importance of the term.

$$\mathrm{idf_{j,k}} = \log \frac{|K|}{1 + \sum_i I(t_k \in p_{i,j})}, \tag{3}$$

where $|K|$ is the total number of entries and $I(\cdot)$ is the indicator function.

Then the element $d_{i,j,k}$ of tensor $\mathcal{D}$ is

$$d_{i,j,k} = \mathrm{tf_{i,j,k}} \times \mathrm{idf_{j,k}}. \tag{4}$$

4.3 Latent Semantic Tensor Indexing

For the CQA tensor, we can decompose it as illustrated in Figure 2.

$$\mathcal{D} = \mathcal{Z} \times_1 \mathbf{U}_{\mathrm{Entry}} \times_2 \mathbf{U}_{\mathrm{Part}} \times_3 \mathbf{U}_{\mathrm{Term}}, \tag{5}$$

where $\mathbf{U}_{\mathrm{Entry}}$, $\mathbf{U}_{\mathrm{Part}}$ and $\mathbf{U}_{\mathrm{Term}}$ are left singular matrices of corresponding flattened matrices. $\mathbf{U}_{\mathrm{Term}}$ spans the term space, and we just use the vectors corresponding to the $1,000$ largest singular values in this paper, denoted as $\mathbf{U'}_{\mathrm{Term}}$.

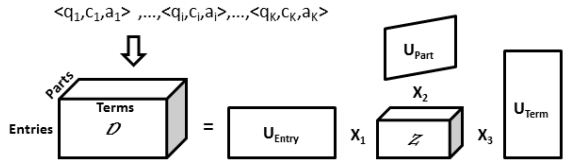

Figure 2: 3-mode SVD of CQA tensor

To deal with such a huge sparse data set, we use singular value decomposition (SVD) implemented in Apache Mahout[3] machine learning library, which is implemented on top of Apache Hadoop[4] using the map/reduce paradigm and scalable to reasonably large data sets.

[3] http://mahout.apache.org/

[4] http://hadoop.apache.org

4.4 Question Retrieval

In order to retrieve similar question effectively, we project each CQA triple $\mathcal{D}_q \in \mathbb{R}^{1\times 3\times T}$ to the term space by

$$\hat{\mathcal{D}}_i = \mathcal{D}_i \times_3 \mathbf{U'}^T_{\mathrm{Term}}. \tag{6}$$

Given a new question only with title part, we can represent it by tensor $\mathcal{D}_q \in \mathbb{R}^{1\times 3\times T}$, and its $\mathbf{M_{Content}}$ and $\mathbf{M_{Answer}}$ are zero matrices. Then we project $\mathcal{D}_q$ to the term space and get $\hat{\mathcal{D}}_q$.

Here, $\hat{\mathcal{D}}_q$ and $\hat{\mathcal{D}}_i$ are degraded tensors and can be regarded as matrices. Thus, we can calculate the similarity between $\hat{\mathcal{D}}_q$ and $\hat{\mathcal{D}}_i$ with normalized Frobenius inner product.

For two matrices A and B, the Frobenius inner product, indicated as $A : B$, is the component-wise inner product of two matrices as though they are vectors.

$$A : B = \sum_{i,j} A_{i,j} B_{i,j} \tag{7}$$

To reduce the affect of length, we use the normalized Frobenius inner product.

$$\overline{A : B} = \frac{A : B}{\sqrt{A : A} \times \sqrt{B : B}} \tag{8}$$

While given a new question both with title and content parts, $\mathbf{M_{Content}}$ is not a zero matrix and could be also employed in the question retrieval process. A simple strategy is to sum up the scores of two parts.

5 Experiments

5.1 Datasets

We collected the resolved CQA triples from the "computer" category of Yahoo! Answers and Baidu Zhidao websites. We just selected the resolved questions that already have been given their best answers. The CQA triples are preprocessed with stopwords removal (Chinese sentences are segmented into words in advance by FudanNLP toolkit(Qiu et al., 2013)).

In order to evaluate our retrieval system, we divide our dataset into two parts. The first part is used as training dataset; the rest is used as test dataset for evaluation. The datasets are shown in Table 2.

DataSet	training data size	test data size
Baidu Zhidao	423k	1000
Yahoo! Answers	300k	1000

Table 2: Statistics of Collected Datasets

Methods	MAP
Okapi	0.359
LSI	0.387
(Jeon et al., 2005b)	0.372
(Xue et al., 2008)	0.381
LSTI	0.415

Table 3: Retrieval Performance on Dataset from Yahoo! Answers

5.2 Evaluation

We compare our method with two baseline methods: Okapi BM25 and LSI and two state-of-the-art methods: (Jeon et al., 2005b)(Xue et al., 2008). In LSI, we regard each triple as a single document. Three annotators are involved in the evaluation process. Given a returned result, two annotators are asked to label it with "relevant" or "irrelevant". If an annotator considers the returned result semantically equivalent to the queried question, he labels it as "relevant"; otherwise, it is labeled as "irrelevant". If a conflict happens, the third annotator will make the final judgement.

We use **mean average precision** (MAP) to evaluate the effectiveness of each method.

The experiment results are illustrated in Table 3 and 4, which show that our method outperforms the others on both datasets.

The primary reason is that we incorporate the content of the question body and the answer parts into the process of question retrieval, which should provide additional relevance information. Different to the translation-based methods, our method can capture the mapping relations in three parts (question, content and answer) simultaneously.

Methods	MAP
Okapi	0.423
LSI	0.490
(Jeon et al., 2005b)	0.498
(Xue et al., 2008)	0.512
LSTI	0.523

Table 4: Retrieval Performance on Dataset from Baidu Zhidao

It is worth noting that the problem of data sparsity is more crucial for LSTI since the size of a tensor in LSTI is larger than a term-document matrix in LSI. When the size of data is small, LSTI tends to just align the common words and thus cannot find the corresponding relations among the focus words in CQA triples. Therefore, more CQA triples may result in better performance for our method.

6 Conclusion

In this paper, we proposed a novel retrieval approach for community-based QA, called LSTI, which analyzes the CQA triples with naturally tensor representation. LSTI is a unified model and effectively resolves the problem of lexical chasm for question retrieval. For future research, we will extend LSTI to a probabilistic form (Hofmann, 1999) for better scalability and investigate its performance with a larger corpus.

Acknowledgments

We would like to thank the anonymous reviewers for their valuable comments. This work was funded by NSFC (No.61003091 and No.61073069) and 973 Program (No.2010CB327900).

References

M.J. Blooma and J.C. Kurian. 2011. Research issues in community based question answering. In *PACIS 2011 Proceedings*.

R. Burke, K. Hammond, V. Kulyukin, S. Lytinen, N. Tomuro, and S. Schoenberg. 1997. Question answering from frequently asked question files: Experiences with the faq finder system. *AI Magazine*, 18(2):57–66.

Deng Cai, Xiaofei He, and Jiawei Han. 2006. Tensor space model for document analysis. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*.

S. Deerwester, S.T. Dumais, G.W. Furnas, T.K. Landauer, and R. Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391–407.

Huizhong Duan, Yunbo Cao, Chin-Yew Lin, and Yong Yu. 2008. Searching questions by identifying question topic and question focus. In *Proceedings of ACL-08: HLT*, pages 156–164, Columbus, Ohio, June. Association for Computational Linguistics.

T. Hofmann. 1999. Probabilistic latent semantic indexing. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 50–57. ACM Press New York, NY, USA.

J. Jeon, W.B. Croft, and J.H. Lee. 2005a. Finding semantically similar questions based on their answers. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 617–618. ACM.

J. Jeon, W.B. Croft, and J.H. Lee. 2005b. Finding similar questions in large question and answer archives. *Proceedings of the 14th ACM international conference on Information and knowledge management*, pages 84–90.

V. Jijkoun and M. de Rijke. 2005. Retrieving answers from frequently asked questions pages on the web. *Proceedings of the 14th ACM international conference on Information and knowledge management*, pages 76–83.

T.G. Kolda. 2002. Orthogonal tensor decompositions. *SIAM Journal on Matrix Analysis and Applications*, 23(1):243–255.

Xipeng Qiu, Qi Zhang, and Xuanjing Huang. 2013. Fudannlp: A toolkit for chinese natural language processing. In *Proceedings of ACL*.

S. Riezler, A. Vasserman, I. Tsochantaridis, V. Mittal, and Y. Liu. 2007. Statistical machine translation for query expansion in answer retrieval. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.

S.E. Robertson, S. Walker, S. Jones, M.M. Hancock-Beaulieu, and M. Gatford. 1994. Okapi at trec-3. In *TREC*, pages 109–126.

K. Wang, Z. Ming, and T.S. Chua. 2009. A syntactic tree matching approach to finding similar questions in community-based QA services. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 187–194. ACM.

X. Xue, J. Jeon, and W.B. Croft. 2008. Retrieval models for question and answer archives. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 475–482. ACM.

J.M. Ye. 2005. Generalized low rank approximations of matrices. *Mach. Learn.*, 61(1):167–191.

G. Zhou, L. Cai, J. Zhao, and K. Liu. 2011. Phrase-based translation model for question retrieval in community question answer archives. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 653–662. Association for Computational Linguistics.