# Multi-sentence Question Segmentation and Compression for Question Answering

Yixiu Wang[1], Yunfang Wu[1(✉)], and Xueqiang Lv[2]

[1] Key Laboratory of Computational Linguistics, Peking University, MOE, Beijing, China
{labyrinth,wuyf}@pku.edu.cn
[2] Beijing Key Laboratory of Internet Culture and Digital Dissemination Research, Beijing, China
lxq@bistu.edu.cn

**Abstract.** We present a multi-sentence question segmentation strategy for community question answering services to alleviate the complexity of long sentences. We develop a complete scheme and make a solution to complex-question segmentation, including a question detector to extract question sentences, a question compression process to remove duplicate information, and a graph model to segment multi-sentence questions. In the graph model, we train a SVM classifier to compute the initial weight and we calculate the authority of a vertex to guide the propagating. The experimental results show that our method gets a good balance between completeness and redundancy of information, and significantly outperforms state-of-the-art methods.

**Keywords:** Question answering · Question compression · Question segmentation · Complex-question analysis

## 1 Introduction

In a CQA service, users usually have a high tendency to ask multi-sentence questions. The complexity of multi-sentence questions reflects on following three aspects: first, a complete question usually contains several sub-sentences; secondly, inner redundancies exist in sentences; thirdly, the misuse of comma and period marks makes sentence more complicate. Figure 1 shows an example question in a Chinese tourism CQA. In Figure 1, the Description gives a complement to the Title, which consists of five sentences. The Title and Sentence 4 in the question thread are actually expressing the same meaning. In the Description, the Sentence 5 is a polite saying. Sentence 2 serves as a context of Sentence 3, which is a question, and Sentence 4 needs Sentence 1 as a complement.

Complex questions in CQAs are very difficult to analyze by traditional approaches. Multi-sentence question segmentation provide with an effective method to alleviate the complexity of long sentences. Our work develops a complete scheme on complex question segmentation. In Section 2, we provide an overview of our approach. In Section 3, and 4, we describe our approach of initial weight calculation, and the propagation of linking scores, respectively. In Section 5, we report the experimental results. Section 6 gives a comparison with previous work. Finlay Section 7 draws conclusions.


J. Li et al. (Eds.): NLPCC 2015, LNAI 9362, pp. 475–483, 2015.
DOI: 10.1007/978-3-319-25207-0_44

Title: 从马经新加坡回中国需不需要提前获得新加坡的签证? *(Whether should I obtain a visa of Singapore in advance if I return to china via Singapore from Malaysia?)*

Description:

Sentence 1: 我从马来西亚经过新加坡回北京，有马来西亚的旅游签证 *(I return back to Beijing via Singapore from Malaysia, and I own a tourism Visa of Malaysia)*

Sentence 2: 之前看到网上新加坡有过境96小时落地签这个东西，只要有96小时 内离开新加坡的机票就可以获得. *(I have learned from the internet that Singapore used to provide a landing check of 96 hours, as long as a flight leaving Singapore in 96 hours is processed.*)

Sentence 3: 但不知道现在还有没有? *(But whether such policy is still valid now?)*

Sentence 4: 如果没有了，是不是如果要在新加坡转机，就需要提前获得新加坡 的签证了呢? (*If not, then if I want to transit in Singapore, whether I have to obtain a visa of Singapore in advance?)*

Sentence 5: 请各位知道的驴友们回答一下，谢谢! *(Looking forward to your answers, thank you!)*

**Fig. 1.** An example question in tourism CQA

## 2 Overview of Approach

Our method transfers the question thread into a directed graph (V, E) in the following steps shown in Figure 2.

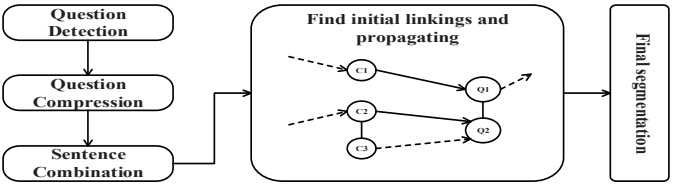


**Fig. 2.** A diagram view of our approach

In our method, the elementary unit (EU) to cope with is a sub-sentence segmented by a comma, instead of a full sentence denoted by a period.

A question detector is designed to find the question sentences in the question thread. The sentences are divided to two different sets: C (context) and Q (question and command). In addition, the meaningless sentences (greetings, polite sayings) that provide no useful information for answer detection are deleted. We use a rule and regex based question detector, which achieves an F score of 92% on Chinese text.

The question compression removes some duplicate information via similarity calculation. We conduct question compression separately on question sub-sentences and non-question sub-sentences. For each sub-sentence, we calculate its similarity with other sub-sentences and sentences to remove the duplication in the question thread. We consider the following factors in calculating the similarity:

- Extended longest common substring
- Number of same word
- Cosine Similarity

For each sub-sentence, we calculate its similarity with other sub-sentence and natural sentence to compress the question thread. If $Sim(S_1, S_2) > \theta$, then the shorter one will be deleted from the set of vertex. We set θ by experience, which is set to 1.3 in our method. Then we connect the remained continuous non-question sentences in a sentence together to make a longer context.

We then model the thread into a directed acyclic graph, and the initial weight is calculated using a SVM classifier in the building of the DAG. We add a propagation of the weight of edge to the graph. Finally, we split the long question thread into a few short sentences associated with the context of each question.

## 3 Finding Linkings Between Sentences

The question thread is modeled to a directed graph $(V, E)$. We find the relationship between sentences by adding edges between vertexes. An edge $u \to v$ demonstrates that vertex v has dependence on u, or sentence u is the context of sentence v, meaning the information represents by v becomes more complete because of the existence of u.

In our model, both a question sentence, and a non-question sentence, can serve as a context of a question. Notably, we only allow edges of $C \to C$, $C \to Q$ and $Q \to Q$. The edge $Q \to C$ is considered as meaningless, because when people are talking, a question usually does not motivate a context. In addition, if a question $Q_2$ appears later than $Q_1$, only the dependence $Q_1 \to Q_2$ is allowed, as usually the earlier question provides information for the later one. Every edge in our model is considered as directed.

We calculate the weight of edges between each pair of possible combination of edges by exploiting various lexical and structure features:

- **KL-divergence**

Given two sentences u and v, we separately construct unigram language model of u as $M_u$ and v as $M_v$. The KL-divergence between languages $M_u$ and $M_v$ is computed as follow:

$$D_{KL}(M_u \| M_v) = \sum_w p(w|M_u) \log \frac{p(w|M_u)}{p(w|M_v)} \quad (1)$$

KL-divergence shows the difference of the probability distributions of $M_u$ and $M_v$ under the same vector space. The KL-divergence is asymmetry.

- **Correlation**

Given two sentences u and v, the correlation between u and v is defined as the degree of their similarity. We use Word2Vec to calculate the similarity between words.

$$Co(u, v) = \sum_{w_i \in u, w_j \in v} \cos(w_i, w_j) \quad (2)$$

- **Coherence**

Usually, the existence of conjunctions and linking words indicate a relationship between sentences. Some Chinese conjunctions appear in pairs, whose order of appearance suggests the dependence between u and v. The conjunction pairs are found in a mandarin Chinese dictionary.

Other features used in out method are listed as follows:

- *if_same_sen: whether two sub-sentences are in a natural sentence*
- *if_pron: whether any of the two sentences contains a pronoun*
- *if_num: whether any of the two sentences contains a number word*
- *if_ns: whether any of the two sentences contains a location name*
- *if_time: whether any of the two sentences contains a time word*
- *if_v: whether any of the two sentences contains a verb*
- *short_length: the shorter length of the two sentences*
- *ratio: the length ratio of a shorter sentence to a longer sentence*
- *word_pair: whether two sub-sentence contains frequent word pair*

Using the above features, a SVM classifier is trained to calculate the initial weight of edges.

## 4 Propagating the Linking Scores

Our SVM classifier provides an initial weight of edges in the graph $(V, E)$. The initial weight of edges presents a direct relationship between sub-sentences; however, indirect relationship exists between sentences. For example, if $C_1$ is the context of $Q_1$, and $Q_1$ provides information for $Q_2$, then $C_1$ is possibly a context of $Q_2$. In such situations, hidden dependences between $C_1 \to Q_1$, and $Q_1 \to Q_2$ should also be considered, while the weight of SVM classifier cannot directly detect the relationship. Correspondingly, given a $C_1 \to Q_1$, although the weight of the edge E ($C_1 \to Q_1$) is not significant enough for a relationship, but if a vertex $Q_2$ makes chain $C_1 \to Q_2 \to Q_1$ exist, then vertex $Q_2$ should strengthen the potential hidden relationship of $C_1 \to Q_1$.

In our propagating method (shown in Figure 3), the weight of $C_1 \to Q_1$ is possibly refreshed by the existence of another vertex V, where $C_1 \to V$, $V \to Q_1$ exist. Whether the refreshing of edge $C_1 \to Q_1$ is efficient is determined by the significant degree of V. Thus, the authority of vertex V is crucial in propagating. We adopt an algorithm similar with page-rank to calculate the authority for each vertex.

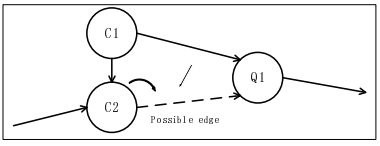


**Fig. 3.** Example of weight propagating

### 4.1 Calculating the Authority

The authority of vertex in the entire question graph is associated with the sematic information provided by the vertex, and the position information of the vertex in the

graph. Our algorithm calculates the in-Degree, out-Degree as the position information, and a High-Frequency salient word count as a measure of the sematic information.

The in-Degree measure of a vertex presents the degree of independence of a vertex. A vertex that owns a high in-Degree refers to a processing of more context, and the owning of more contexts indicate that the vertex itself provides less information for others. Therefore, the in-Degree measure is a negative factor for the authority of the vertex. In-Degree is computed by:

$$in - Degree(v) = e^{-\lambda \times \frac{in(v)}{|V|}} \tag{3}$$

The out-Degree measure of a vertex shows the importance of the vertex. A vertex with high out-Degree indicates the vertex is considered as the context of many other vertexes. Out-Degree is computed by:

$$out - Degree(v) = \frac{out(v)}{|V|} \tag{4}$$

In a question, some salient words carry very important information for answer detection. We count the ratio of salient words in each sentence as $salient(v)$.

The authority of a vertex is calculated as follow:

$$Au(v) = \alpha In - Degree(v) + \beta Out - Degree(v) + \gamma Salient(v) \tag{5}$$

$\alpha, \beta, \gamma$ are set to 0.3, 0.3 and 0.4. We add a normalization method like PageRank algorithm. When computing the authority of v, we take all the valid generators of v into consideration:

$$Au(v) = \sum_{u \in G(v)} Au(u) \times e^{\mu \times weight(u,v)} \tag{6}$$

(6) shows that the generators of v's contribution to v differed by weight (u, v). As the weight is only the initial weight, we give a very small $\mu$ to reduce the influence of weight. We suppose the propagating chain to be brief and short, thus the PageRank–like algorithm stops after the first layer. We add final normalization as following:

$$Au(v) = |V| \times \frac{Au(v)}{\sum_{u \in V} Au(u)} \tag{7}$$

### 4.2 Propagating the Scores

Our propagating method refreshes the weight in the following algorithm:

$$w(u,v) \to \frac{w(u,v)}{2} + \lambda Au(a)\sqrt{w(u,a) * w(a,v)} \tag{8}$$

We only refresh $w(u,v)$ when it becomes larger. $\lambda$ is a damping factor to reduce the propagating chain from growing longer. We repeat the calculation until no changes of weight occurs in the graph.

### 4.3 Getting Final Segmentation

We use a dynamic algorithm to determine whether the relationship exists between two sentences. The edges are sorted in descending order. We successively deal with each edge $E_i$, and the algorithm stops if the $weight_i$ is below 0.5, or

$$weight_{i-1} - weight_i > \frac{1.5(weight_1 - weight_i)}{i} \tag{9}$$

## 5 Experiments

Our data comes from XieCheng tourism forum, which is a famous traveling CQA forum in China. We randomly choose 1,200 sentences with more than five sub-sentences as our training data. An under-graduate student annually annotated the direct relation between contexts and questions. We train our Word2Vec on a 100M tourism corpus. We train a SVM classifier to compute an initial weight of the relations (Section 3), and then we add a further propagation to refresh the closeness scores (Section 4).

We evaluated the effectiveness of our methods by user tests. We proposed an evaluation metric via four different aspects:

- *Total redundancy (TR):, which shows whether the segmentation results have redundancy information between different segments.(1 point)*
- *Total completeness (TC), which shows whether the segmentation results present all the question information in the initial question thread.(1 point)*
- *Segment redundancy (SR), which shows the degree of the appearance of un-related information in each segment.(2 points)*
- *Segment completeness (SC), which shows whether each question segment is complete and explicit to find answers.(2 points)*

In our evaluation metric, redundancy and completeness are complementary. When a segmentation result has a high degree of completeness, usually it has a high tendency of containing more information that is redundant.

We set up two kinds of baselines. First, we use the natural sentence segmentation (NSS) as a baseline, which use the question detector to find the question sub-sentence, and then cut the question sentence by the full stop as the segmentation result. Second, we employ the algorithm of MQS [6] as our baseline. Our method in this paper is denoted as Compression and Propagation Segmentation (CPS).

We randomly choose 200 sentences outside the training data in the same CQA forum as the test data. Three systems (NSS, MQS, and CPS) are employed to run on the test data to get three different segmentation results. Finally, according to the above-mentioned metrics, two human annotators are required to give points to each segmentation result of three methods, without knowing which system generates the result. The evaluation results are shown in Table 1.

From Table 1, our system CPS obtains the best subjective performance by both annotators. Our method achieves an average total point of 4.896 (0.000~6.000), which

considerably outperforms the other two baseline systems. The simple NNS method rivals the previous MQS approach in total score.

Comparing with NNS method, our algorithm notably raises the point of SC, with only a small decrease in the other three metrics. On the other hand, our algorithm CPS significantly outperforms the previous MQS approach both in TR and in SR, with only a little drop in CT and SC. The previous approach MQS has a high tendency to joint short sentences to a longer one, so the system is likely to cover more information and shows better completeness but resulting in much redundancy. Our method finds a better way to balance the completeness and redundancy, and the experimental results demonstrate its effectiveness.

**Table 1.** Evaluation results of Segmentation

| Systems | | TR | TC | SR | SC | Total |
|---|---|---|---|---|---|---|
| NNS | Tester 1 | 0.839 | 0.964 | 1.668 | 1.197 | 4.668 |
| | Tester 2 | 0.865 | 0.99 | 1.616 | 1.161 | 4.601 |
| MQS | Tester 1 | 0.275 | 0.984 | 0.980 | 1.766 | 4.005 |
| | Tester 2 | 0.269 | 0.979 | 0.102 | 1.792 | 4.062 |
| CPS | Tester 1 | 0.777 | 0.881 | 1.554 | 1.684 | **4.896** |
| | Tester 2 | 0.746 | 0.891 | 1.518 | 1.632 | **4.777** |

## 6 Related Work

Sentence segmentation is a basic method to alleviate the complexity of multi-sentence questions. Many previous researches have studied on the direction of long sentence segmentation, and the methods can be divided into two genres: chunking-based sentence segmentation and segmentation based on finding relations. The chunking may use rule and regex method [1], decision tree [2], a maximum entropy model, and conditional random field (CRF) [3] have been proposed to deal with segmentation. Takechi [3] proposed a method of combining unigram and bigram word features to reach better segmentation result. The question segmentation has a strong dependence on question detection, and the question detection can be vector space model [3], language model [4], translation model [5], syntactic tree matching model [6] and the recently proposed convolutional neural network model [7]. However, in CQAs, most sentences are long and complex, and the context of a question is often not adjacent to the question itself, which make chunking-based methods less effective.

Among segmentation based on finding relations, Wang et al. (2010) [8] address the problem of multi-sentence question segmentation towards further analysis of question sentences. They build a question detector to extract question sentences and context sentences, and propose a simple graph based approach to segment multi-sentence questions with simple propagating method.

Wang's technique and ours differ in the following points: our method provides a robust algorithm of finding semantic relations between sentence by expanding word2vector, and use synonyms to deal with lexical gap; we exploit an effective propagation method with an authority-calculating algorithm, which reduce the redundancy of

propagating significantly. Many studies worked on calculating the authority of vertex in an acyclic graph to find crucial vertexes [9, 10, 11, 12, 13]. In addition, we add a sentence compression to decrease the duplication in complex CQA sentences.

## 7 Conclusion

In this paper, we propose a robust method to split multi-sentence questions. The sentence compression method effectively reduces the duplication in the question sentence. Our method exploits various lexical and structure features, applies Word2Vec on finding related words, and use synonyms to deal with lexical gap. We also propose an effective propagating method to refresh the relation by a graph algorithm. Our method balances the information completeness and redundancy of multi-sentence segmentation and our result outperform the state-of-art. Our further study will focus on the application of question segmentation on multi-sentences question retrieval, and employ the sentence segmentation on complex non-question sentences.

**Acknowledgement.** This work is supported by Humanity and Social Science foundation of Ministry of Education (13YJA740060), National High Technology Research and Development Program of China (2015AA015403), Key Program of Social Science foundation of China (12&ZD227), and the Opening Project of Beijing Key Laboratory of Internet Culture and Digital Dissemination Research (ICDD201302).

## References

1. Ang, J., Liu, Y., Shriberg, E.: Automatic dialog act segmentation and classification in multiparty meetings. In: Proc. Int. Conf. Acoust Speech, Signal Process (2005)
2. Takechi, M., Tokunaga, T., Matsumoto, Y.: Chunking-based question type identification for multi-sentence queries. In: SIGIR (2007)
3. Duan, H., Cao, Y., Lin, C.-Y., Yu, Y.: Searching questions by identifying question topic and question focus. In: HLT-ACL (2008)
4. Jeon, J., Croft, W.B., Lee, J.H.: Finding similar questions in large question and answer archives. In: CIKM (2005)
5. Riezler, S., Vasserman, A., Tsochantaridis, I., Mittal, V., Liu, Y.: Statistical machine translation for query expansion in answer retrieval. In: ACL (2007)
6. Wang, K., Ming, Z., Chua, T.-S.: A syntactic tree matching approach to finding similar questions in community-based qa services. In: SIGIR (2009)
7. Yinh, W., He, X., Meek, C.: Semantic parsing for single-relation question answering. In: ACL (2014)
8. Wang, K., Ming, Zh., Hu, X., Chua, T.: Segmentation of multi-sentence questions: towards effective question retrieval in cQA services. In: SIGIR (2010)
9. Cong, G., Wang, L., Lin, C., Song, Y., Sun, Y.: Finding question-answer pairs from online forums. In: SIGIR (2008)
10. Mihalcea, R.: Unsupervised large-vocabulary word sense disambiguation with graph-based algorithms for sequence data labeling. In: EMNLP (2005)

11. Navigli, R., Lapata, M.: Graph connectivity measures for unsupervised word sense disambiguation. In: IJCAI (2007)
12. Sinha, R., Mihalcea, R.: Unsupervised graph-based word sense disambiguation using measures of word semantic similarity. In: ICSC (2007)
13. Hessami, E., Mahmoud, F., Jadidinejad, A.: Unsupervised graph-based word sense disambiguation using lexical relation of WordNet. In: IJCSI (2011)