Tsinghua Science and Technology

Volume 24 | Issue 3 Article 3

2019

Enhanced Answer Selection in CQA Using Multi-Dimensional Features Combination

Hongjie Fan
the School of Electronics Engineering and Computer Science, Peking University, Beijing 100871, China.

Zhiyi Ma
the School of Electronics Engineering and Computer Science, Peking University, Beijing 100871, China.

Hongqiang Li
the School of Software and Microelectronics, Peking University, Beijing 100871, China.

Dongsheng Wang
the School of Software and Microelectronics, Peking University, Beijing 100871, China.

Junfei Liu
National Engineering Research Center for Software Engineering, Peking University, Beijing 100871, China.

TSINGHUA SCIENCE AND TECHNOLOGY
ISSNll1007-0214 09/10 pp346–359
DOI: 10.26599/TST.2018.9010050
Volume 24, Number 3, June 2019

Enhanced Answer Selection in CQA Using Multi-Dimensional Features Combination

Hongjie Fan, Zhiyi Ma*, Hongqiang Li, Dongsheng Wang, and Junfei Liu

Abstract: Community Question Answering (CQA) in web forums, as a classic forum for user communication, provides a large number of high-quality useful answers in comparison with traditional question answering. Development of methods to get good, honest answers according to user questions is a challenging task in natural language processing. Many answers are not associated with the actual problem or shift the subjects, and this usually occurs in relatively long answers. In this paper, we enhance answer selection in CQA using multi-dimensional feature combination and similarity order. We make full use of the information in answers to questions to determine the similarity between questions and answers, and use the text-based description of the answer to determine whether it is a reasonable one. Our work includes two subtasks: (a) classifying answers as good, bad, or potentially associated with a question, and (b) answering YES/NO based on a list of all answers to a question. The experimental results show that our approach is significantly more efficient than the baseline model, and its overall ranking is relatively high in comparison with that of other models.

Key words: community question answering; information retrieval; multi-dimensional features extraction; similarity computation

1 Introduction

Web forums for Community Question Answering (CQA), such as Stack Overflow, provide an interface for users to share knowledge, and communicate with each other[1]. CQA is a powerful mechanism that allows users to freely ask questions and look forward to obtaining good, honest answers. In this way, users can obtain specific answers to their questions, instead of searching through large volumes of information.

The increasing popularity of CQA websites has caused number of questions and new forum members to surge without restriction. Unfortunately, much effort must be exerted to assess all possible answers and select one that is the most accurate for a specific question. Thus, although the idea of receiving a direct response to a certain request for information is very appealing, CQA websites also cannot guarantee the quality of the information provided[2]. Many answers are often loosely related to the actual question, and some even shift the topic away from the subject. This issue is a common finding in relatively long answer, i.e., as the answer continues, users begin to discuss with each other instead of answering the initial question.

This issue presents a real problem, as a question can have hundreds of answers. Thus, searching for good answers among the many responses is necessary but time-consuming for participants. Some studies have

• Hongjie Fan and Zhiyi Ma are with the School of Electronics Engineering and Computer Science, Peking University, Beijing 100871, China. E-mail: hjfan@pku.edu.cn; mazhiyi@pku.edu.cn.
• Hongqiang Li and Dongsheng Wang are with the School of Software and Microelectronics, Peking University, Beijing 100871, China. E-mail: hongqiang.li@pku.edu.cn; wangdsh@pku.edu.cn.
• Junfei Liu is with National Engineering Research Center for Software Engineering, Peking University, Beijing 100871, China. E-mail: liujunfei@pku.edu.cn.
∗ To whom correspondence should be addressed.
Manuscript received: 2018-01-10; accepted: 2018-02-15

exploited techniques such as the use of syntactic tree structures to measure the semantic correspondence between a question and answer or translation-based language models for matching by transferring the answer to the corresponding question.

However, guaranteeing the quality of the provided information remains a major challenge. In this work, we adopt the following ideas to tackle the issue of quality in CQA:

(1) Traditional Question Answering (QA) based on similarity sorting. We make full use of answers to questions and find the similarity between them. We then look at the forum closely and speculate that if the questions and answers are similar, then the answer must be more credible. In addition, if questions and answers are presented by the same user, the similarity among them could be higher than between questions and answers provided by different users.

(2) Text-based descriptions of the answer to determine whether the answer is reasonable. We consider that if the answer is long, then it is likely to be a reasonable answer. In addition, if the answer is *Yes*, *No*, and several other words, then it likely to be a reasonable answer. In this work, we take *SemEval-2015 Task 3* (http://alt.qcri.org/semeval2015/) on answer selection in CQA to verify our hypothesis. Two subtasks are described in Section 2.2.

To handle these tasks, we enhance answer selection using the multi-dimensional feature combination method. Our main contributions are as follows:

(1) We present a framework for enhancing answer selection in CQA using the multi-dimensional feature combination method.

(2) We extract information for each question and comment from the data set. Twenty features are extracted based on content description, text similarity, and attribute description.

(3) We build a model from these features using the machine learning approaches, Support Vector Machine (SVM), Gradient Boosting Decision Tree (GBDT), and random forest, to classify the dimensions obtained.

(4) We conduct an experiment to show that our three approaches are significantly more efficient than the baseline model, and that its overall ranking is relatively high compared with those of other methods.

The rest of this paper is organized as follows: Section 2 introduces the background and preliminaries of CQA, task definition, and multi-dimensional feature extraction. Section 3 presents the methods including the selection of features, formulation of labels, and construction of these models. Experiments and the related discussion are conducted and provided in Sections 4 and 5, respectively. Section 6 analyzes the related work, and Section 7 draws conclusions and offers directions for future work.

2 Background

In this section, we briefly introduce the concepts of CQA, task description, and multi-dimensional feature extraction.

2.1 CQA

CQA websites provide an interface for users to share knowledge[3]. Unfortunately, given the increasing popularity of CQA forums, a user must browse all possible answers and understand them before arriving at the correct answer. Answer selection in CQA recognizes high-quality responses, which is greatly valuable for information retrieval, and aims to automatically categorize answers as *good* if they completely answer the question, *bad* if they are irrelevant to the question, and *potential* if they contain useful information about the question but do not completely answer it. Figure 1 shows an example question and its four possible answers. The dashed arrows depict the relationships of the answers in the sequence.

The title of *"Can we resign from the job"* gives a brief summary of the question, and the body describes the question in detail. The question includes four parts. After checking all answers, good answers among a1, a2, a3, and a4 can be obtained. Answer a1 clearly is a *good* answer because it provides helpful information such as *"contact"*, and *"require the employee to pay a certain amount of fee"*. Besides, this answer mentions

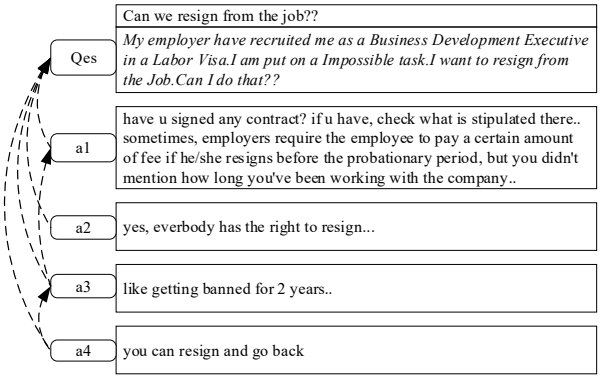


Fig. 1 An example of the answer body for a question.

other details such as *"how long have you been working with the company"*. This answer is also long, which means it is likely to be a reasonable answer. Answer a2 is also a reply to the question but it does not contain any useful information; thus, it is regarded as a *bad* answer. In answer a4, some sentences are auxiliary and do not provide meaningful information for answer selection in CQA, it is better than answer a2 but not good enough as an answer result for the question. As such, a4 can be treated as a *potential* answer. Some answers, such as a3 and a4, would reply the former answer.

When a large number of CQA websites are available, a CQA website must be able to provide high-quality content to distinguish itself from other websites. Thus, although the idea of receiving a direct response to a certain need sounds very appealing, some risk is present because the quality of the provided information cannot be guaranteed[3].

In Ref. [4], the authors show a correlation between question quality and answer quality; *good* answers are more likely to be given in response to *good* questions, while *bad* answers appear in response to *bad* questions. According to the definition in Ref. [5], higher quality questions can be expected to draw greater user attention, include more answer attempts, and obtain the best answer within a shorter period of time than poorer-quality questions.

2.2 Task definition

Based on this description of CQA, we take the *SemEval-2015 Task 3* in CQA to test our hypotheses. This task includes two subtasks:

Task A asks types of prediction for all questions under given questions and answers. It gives a question (including a short title and extended description) and its answers and then divides each answer into one of the following:

(a) Absolutely related (*good*);

(b) Bad or irrelevant (*bad*, *dialog*, *non-English*, or *other*) or;

(c) Potentially useful (*potential*).

Task B asks the prediction about *Yes*, *No*, or *Unsure* based on a list of all answers. It gives a *YES/NO* type problem (including a short title and its extended description) and a few answers based on good answers to the questions asked in task A.

2.3 Multi-dimensional feature extraction

Features are extracted from the answers, and feature values are passed through a regression model. Thus, we need to assign a regression value to each answer quality class. For example, a regression value of 1.0 can be assigned to *good* answers, a value of 0.5 can be assigned to *potential* answers, and a value of 0.0 can be assigned to *bad* ones.

Multi-dimensional feature extraction includes two parts: features based on contents and features based on attributes. We provide details for each type of feature extraction in Section 3.2. Five similarity calculation models, including Latent Semantic Index (LSI)[6], Latent Dirichlet Allocation (LDA)[7], Senctence LDA (SenLDA)[8], Word2Vector[9], and BagofWords[10], are proposed to analyze the final output answer quality scores.

3 Framework of Enhanced Answer Selection in CQA

For processing the enhanced answer selection in CQA, we combine multi-dimensional features to build several models. In this section, we present the process of model-building, which includes three components as shown in Fig. 2: data preprocessing, types of feature extraction, and classification of all features using SVM, GBDT, and random forest.

In the first step, data preprocessing is conducted to input data into the storage system. The parsing and partitioning steps are normally executed once from the given document. In the second step, we extract features based on contents and attributes. In the third step, we build models using the extraction features and assign an answer quality score.

3.1 Data preprocessing

Data preprocessing is an important phase of model-building as it extracts various information from websites and presents it in a database.

As shown in Fig. 3, an XML parser receives the input corpus in XML format. The XML file contains all of the questions, along with their respective answers.

An XML interpreter extracts the questions and associated answers. Here we describe the main preprocessing steps on a collection of CQA-QL corpus of *SemEval-2015 Task 3* on answer selection.

During data preprocessing, tokenization, stop word removal, and stemming are common tasks applied to process the content of the documents. First, we tokenize

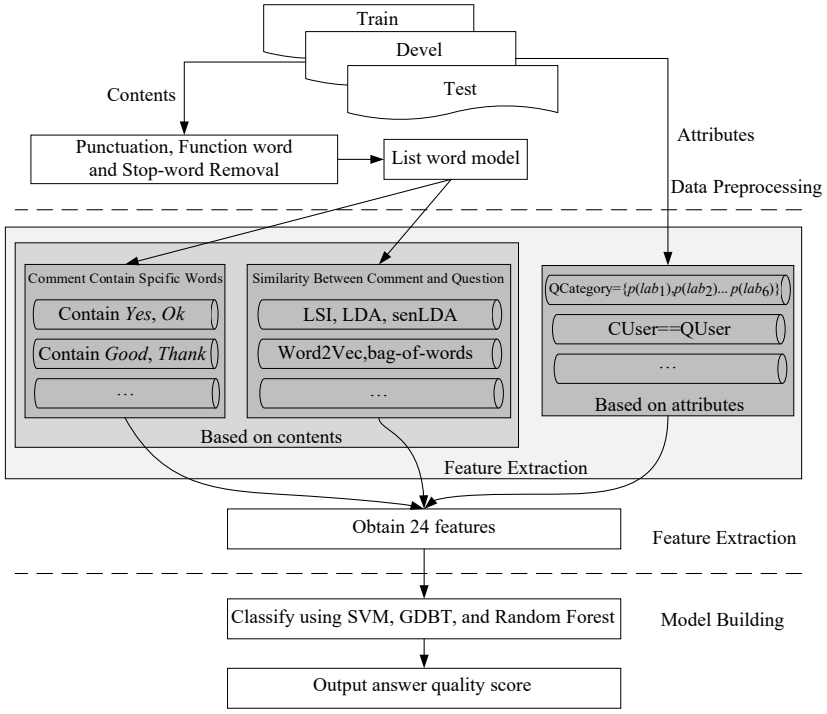


Fig. 2 Framework of enhanced answer selection in CQA.

```
<Question QID="Q7" QCATEGORY="Visas and Permits" QDATE="2010-04-28 05:57:26" QUSERID="U22" QTYPE="YES_NO" QGOLD_YN="Yes">
        <QSubject>Can we resign from the job??</QSubject>
        <QBody>My employer have recruited me as a Business Development Executive in a Labor Visa.I am put on a Impossible
task.I want to resign from the Job.Can I do that??</QBody>
        <Comment CID="Q7_C1" CUSERID="U23" CGOLD="Good" CGOLD_YN="Unsure">
                <CSubject>have u signed any contract?</CSubject>
                <CBody>have u signed any contract? if u have, check what is stipulated there.. sometimes, employers requir
e the employee to pay a certain amount of fee if he/she resigns before the probationary period, but you didn't mention how
 long you've been working with the company..</CBody>
        </Comment>
        <Comment CID="Q7_C2" CUSERID="U24" CGOLD="Good" CGOLD_YN="Yes">
                <CSubject>yes, everbody has the right to resign...</CSubject>
                <CBody>the problem is...expenses</CBody>
        </Comment>
        <Comment CID="Q7_C3" CUSERID="U25" CGOLD="Dialogue" CGOLD_YN="Not Applicable">
                <CSubject>like getting banned for 2</CSubject>
                <CBody>like getting banned for 2 years&gt; Is that right?</CBody>
        </Comment>
        <Comment CID="Q7_C4" CUSERID="U26" CGOLD="Good" CGOLD_YN="Yes">
                <CSubject>you can resign and go back</CSubject>
                <CBody>you can resign and go back</CBody>
        </Comment>
</Question>
```

Fig. 3 A sample example from the CQA-QL corpus.

articles with the help of the Python Natural Language Toolkit (NLTK)[11] and a set of predefined regular expressions. NLTK is a leading platform used to allow Python programs to work with human language data through easy-to-use interfaces with over 50 corpora and lexical resources. Therefore, it has been called "a wonderful tool for dealing computational linguistics", and "an amazing library to play with natural language". Then, we standardize the tokens by removing noise and stop-words from the list of words; words such as *the*, *on*, and *in* were also removed. We use typical normalization techniques by utilizing a stemmer algorithm that transforms a word into its original form. After building a vocabulary of corpus words, each document is represented as a sparse "bag of words". All of the text from the cleaned documents are split into words. Finally, we used the processed documents as inputs to the algorithm, which will learn the structure of the corpus from the word frequencies of the corresponding documents[12].

After data preprocessing, the content and attribute information is processed as training data, devel data, and test data.

3.2 Types of feature extraction

XML file cannot be split because they include opening and closing tags at the beginning and end, respectively, of the document. As such, we cannot begin processing at any point between those tags.

For example, we have an XML document representing Fig. 2, which includes the text "⟨*QSubject*⟩ *Can we resign from the job??*⟨*/QSubject*⟩". A fatal

error is produced in XML grammar if the document is split into fs_i representing "⟨*QSubject*⟩*Can we resign from the job??*⟨/" and fs_{i+1} representing "*QSubject*⟩", because of the end-tag of an element must be intact in the form of ⟨/*'tagname'*⟩.

To solve this problem, the raw data can be divided into two parts: the content and the attribute of the XML information. The content information includes the content description, so as the text in specific tags such as *QSubject*, *QBody*, *CSubject*, and *CBody*. The attribute information includes tags such as *QID*, *QCATEGORY*, *QUSERID*, *QTYPE*, *QGOLD_YN*, *CID*, *CUSERID*, *CGOLD*, and *CGOLD_YN*.

After preprocessing the raw data, we classify features into three types according to their characteristics: based on content description, based on text similarity, and based on attribute description.

The feature types, descriptions, and feature numbers are presented in Table 1.

3.2.1 Feature extraction based on content description

We have selected some key words to describe whether the answer is reasonable or not. Table 2 shows some of these features (13 features in this example) based on content description.

For example, if the answer to the text description appears "*URL*", "*Email*", which represents the key words of information, we think that the answer may contain valid information, which is a reasonable answer.

Table 1 Types of feature extraction.

Feature type	Description	Number of features
Type1	Based on content description	13
Type2	Based on text similarity	5
Type3	Based on attribute description	2

Table 2 Features based on content description.

Attribute	Description	Dimension	Value type
hasURL	Has(Not) URL	1	Bool
hasYes	Number of Yes in Contents	1	Int
hasOk	Number of Ok/Okay	1	Int
...	...	...	...
startWithYes	startWithYes or Not	1	Bool
wordNums	Number of Words	1	Int

If *Yes*, *No*, *OK*, and similar responses are provided, the answer could be the effective answer to the question. In addition, if the length of the answer is long, the information included in the answer information could be substantial and the answer is likely to be valid.

3.2.2 Feature extraction based on text similarity

Meaningful words can express the information of an entire sentence. Therefore, we determine the answer to a question through text similarity. We choose five popular similarity calculation models, namely LSI[6], LDA[7], SenLDA[8], Word2Vector[9], and BagofWords[10], to analyze the final output answer quality score. These models are unsupervised.

Table 3 shows all similarity calculation models based on text similarity. These models can express the semantics of the whole text in various forms. Thus, they must be built with the training, devel, and test datasets as inputs for training. Finally, we obtain the similarity measure of two texts by calculating the cosine distance of their vectors.

(1) LSI-based feature extraction

LSI[6] is an effective spectral document clustering method. This model explains text documents by mixing latent topics and analyzes the relationships between documents and terms. LSI decomposes the text vector onto the vector space of the topic size of the dimension through singular value decomposition to identify patterns in the relationships between the terms and concepts contained in a collection of texts.

A key feature of LSI is its ability to retain latent structures in words; thus, it improves the clustering efficiency. Using Singular Value Decomposition (SVD), we project the text word from the three corpora to the dimensional vector space. Thereafter, we compute the *tf-idf* value of each word in the documents.

(2) LDA-based features extraction

Topic models are a set of models representing documents in a collection or corpus. They enable us to represent the properties of a large corpus containing numerous words with a small set of topics by

Table 3 Features based on text similarity.

Attribute	Description	Dimension	Value type
Sim_{LSI}	LSI_Sim (Q, C)	1	Float
Sim_{LDA}	LDA_Sim (Q, C)	1	Float
$\text{Sim}_{\text{SenLDA}}$	SenLDA_Sim (Q, C)	1	Float
$\text{Sim}_{\text{W2Vec}}$	W2Vec_Sim (Q, C)	1	Float
Sim_{BOWS}	Bows_Sim (Q, C)	1	Float

representing documents according to these topics. Blei et al.[7] proposed LDA as a generative probability model of a corpus that can be used to estimate multinomial observations using unsupervised learning.

A probabilistic generative model considers data as observations originating from a generative probabilistic process that includes hidden variables. The hidden variable is typically inferred via posterior inference. In posterior inference, one tries to identify the posterior distribution of the hidden variables that are conditioned on the observations.

LDA can set the number of topics to be determined, and the topic distribution of each text is obtained by training. Table 4 shows notations for LDA where the words w of a document share the same topic z and SenLDA (details to be presented later).

Topics in LDA are drawn from a conjugate Dirichlet prior that remains the same for all documents. For each document d in a corpus D, LDA assumes the following generative process (Fig. 4):

(a) Sample document length N_d from a Poisson distribution $N_d \sim Poisson\,(\xi)$.

Table 4 Notations for LDA and SenLDA.

Symbol	Description
D	Document corpus.
S_d	Number of sample sentence.
N_d	Number of words.
Z	All topics.
W	All words.
$z_{d,n}$	Topic of the n-th word in the d-th document.
$z_{n,s}$	Topic of the s-th word in the n-th sentence.
$w_{d,n}$	n-th word in the d-th document.
$w_{d,s}$	s-th word in the d-th sentence.
$\vec{\theta}_d$	Distribution of topics specific to document d.
$\vec{\varphi}_k$	Distribution of words specific to topic k.
α	Hyper-parameter of the topic distribution $\vec{\theta}_d$.
β	Hyper-parameter of the topic distribution $\vec{\varphi}_k$.

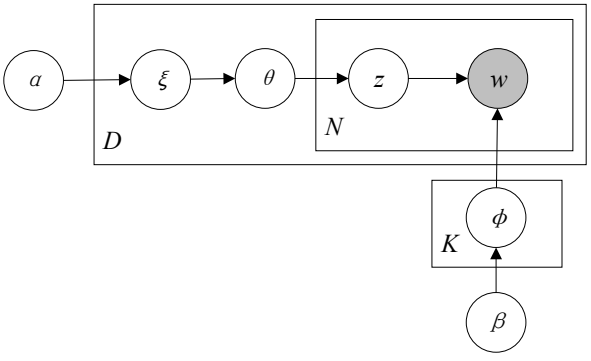


Fig. 4 LDA model.

(b) Pick a topic distribution $\vec{\theta}_d$ from a Dirichlet distribution $\vec{\theta}_d \sim Dirichelet\,(\alpha)$.

(c) For the n-th word in N_d words:

(i) Choose a topic $z_{d,n}$ from the multinomial distribution $z_{d,n} \sim Multinomial(\vec{\theta}_d)$.

(ii) Choose a word $w_{d,n}$ from the multinomial distribution $Multinomial(\vec{\varphi}_{z_{d,n}})$ with the topic-word distribution $\vec{\varphi}_{z_{d,n}}$ sampled from a Dirichlet distribution $\vec{\varphi}_{z_{d,n}} \sim Dirichelet(\beta)$.

LDA estimation is achieved using the approximate estimation method called Gibbs sampling[13]. We sample the topic assignment of each word w following the multinomial distribution:

$$p(z_w = k|\vec{z}_{\neg w}, \vec{w}, \alpha, \beta) = \frac{n_{k,\neg w} + \beta}{\sum_{v=1}^{|\vec{w}|}(n_{k,v} + \beta) - 1} \frac{n_{d,\neg w} + \alpha}{\sum_{j=1}^{K}(n_d^j + \alpha) - 1} \quad (1)$$

where $\vec{z}_{\neg w}$ denotes the topic assignments of all words except the current assignment, $n_{k,\neg w}$ is the number of times topic k is assigned to the word w except in the current assignment, $\sum_{v=1}^{|\vec{w}|} n_{k,v} - 1$ is the total number of times that topic k is assigned to words in vocabulary $n_{d,\neg w}^k$ except the current assignment, $\vec{w}$ is the number of words in document d assigned to topic k except the current assignment, and $\sum_{j=1}^{K} n_d^j - 1$ is the total number of words in document d, not counting the current word.

At the last iteration of sampling, the topic assignment of each word is saved to enrich the corpus. Traditionally, topic models assume that each word occurrence within a document is independent.

(3) SenLDA-based feature extraction

SenLDA extends the LDA by adding an extra plate denoting the coherent text segments of a document. The goal of SenLDA is to overcome the limitation of LDA by incorporating the structure of the text in the generative and inference processes.

As shown in Fig. 5, SenLDA assumes the following

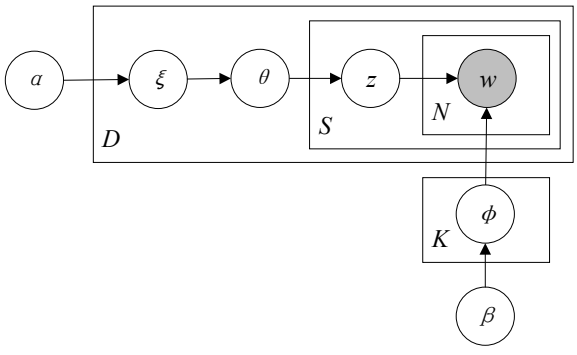


Fig. 5 SenLDA model.

generative process:

(a) Sample sentence number S_d from a Poisson distribution $S_d \sim Poisson(\xi)$.

(b) Pick a topic distribution $\vec{\theta}_d$ from a Dirichlet distribution $\vec{\theta}_d \sim Dirichelet(\alpha)$.

(c) For the n-th word in N_d words:

(i) Sample a number of words from a Poisson distribution $W_d \sim Poisson(\xi)$.

(ii) Choose a sample topic $z_{d,s}$ from the multinomial distribution $z_{d,s} \sim Multinomial(\vec{\theta}_d)$.

(d) Choose words w in the $w_{d,s}$ from a Dirichlet distribution $w \sim Dirichelet(\phi_{z_{d,s}})$.

SenLDA parameters are also estimated using Gibbs sampling[13]. We sample the topic assignment of each word w in each sentence following the distribution:

$$p(z_s = k|\vec{z}_{\neg s}, \vec{w}) = (n^{(k)}_{m,\neg s} + \alpha)\times \frac{\prod_{w\in s}(n^{(w)}_{k,\neg s} + \beta)...((n^{(w)}_{k,\neg s} + \beta) + (n^{(w)}_{k,s} - 1))}{(\sum_{w\in V}(n^{(w)}_{k,\neg s} + \beta))...(\sum_{w\in V}(n^{(w)}_{k,\neg s} + \beta) + (n^{(w)}_{k,s} - 1))} \quad (2)$$

where $\vec{z}_{\neg s}$ denotes the topic assignments of all sentences except the current assignment, $n_{k,\neg s}$ is the number of times topic k assigned to the sentences except in the current assignment, and $n^{(w)}_{k,\neg s}$ denotes the number of times that topic k is observed in sentences from document d, excluding the sentence currently being sampled. The number of words in document d is assigned to topic k.

We use the LDA and SenLDA models as topic model-based features to transform questions and answers into topic vectors and calculate the cosine similarity between the topic vectors of the question and its answers. After experimenting on the development set, the LDA and SenLDA models built from the training data are considered effective and, thus, used in the final system.

(4) Word2Vec-based feature extraction

Word2Vec[9] uses a simple neural network architecture consisting of an input layer, a projection layer, and an output layer to generate high-dimensional vector representations for each word or document that can predict nearby words well. The Word2Vec model can train the vector representation of each word, and the word vector representation of the text is obtained by summing all of the words of the whole text.

We use Word Vector representation-based features to model the relevance between the question and its answer. All of the questions and answers are tokenized, and the words are transformed into vector using the pretrained Word2Vec model. Each word in the question will then be aligned with the word in the answer that has the highest vector cosine similarity. The returned value will be the sum of the scores of these alignments normalized by the question's length:

$$align(w_i) = \max_{0<j\leqslant m}(\cos(w_i, w'_j)) \quad (3)$$

$$sim_{word2vec} = \sum\nolimits_{i=1}^{n} align(w_i)/n \quad (4)$$

where $\cos(w_i, w'_j)$ is the cosine similarity of two vector representations of the i-th word in the question with the j-th word in the answer, and n and m are the lengths (in number of words) of the question and answer, respectively. Here, we use cosine distance since, compared with other distances, such as Euclidean distance, the cosine distance pays more attention to the difference of two vectors.

(5) BagofWords-based feature extraction

The BagofWords (BOW) model[10] is a recently proposed framework for learning continuous word representations based on the distributional hypothesis. The BagofWords representation disregards the linguistic structures between words. Each dimension represents the frequency of a word, and the cosine distance is used to measure the similarity of two texts. We learn parameter values to maximize the log likelihood of each token given its context:

$$L(\theta) = \frac{1}{N}\sum\nolimits_{i=1}^{N}\log p(w_i|w_{i-k}^{i+k}) \quad (5)$$

where N is the size of the corpus and w_{i-k}^{i+k} is the set of words in the window of size k centered at w_i (w_i excluded). The BOW model formulates the probability $p(w_i|w_{i-k}^{i+k})$ using a softmax function as follows:

$$p(w_i|w_{i-k}^{i+k}) = \frac{\exp(v'_{w_i}\cdot\sum_{-k\leqslant j\leqslant k, j\neq 0} v_{w_{i+j}})}{\sum_{w\in W}\exp(v'_{w_i}\cdot\sum_{-k\leqslant j\leqslant k, j\neq 0} v_{w_{i+j}})} \quad (6)$$

where v_w and v'_w represent the input and output vectors of the word w, respectively. To train the model efficiently, hierarchical softmax and negative sampling techniques are used[9]. Some morphology-based methods have been proposed, for example, Ref. [14].

BagofWords features include n-grams, parts of speech, and features that account for the presence and absence of subjective words. The approach is quite simple to implement and attractive because this model reduces the feature space of a potentially large number of dimensions and can help classifiers boost their

performance[15,16].

3.2.3 Feature extraction based on attribute description

In terms of question and answer attribute information, if the problem user and the answer user are the same person, then the question is related to the answer. Different types of questions may be difficult to answer because of the different contents involved. Based on this observation, we extract the features shown in Table 5.

3.3 Model building

Classification is a general process related to categorization, the process through which objects are differentiated. In this paper, the types of features we have obtained must be classified. The common classification models include logistic regression, SVM, naive Bayes, GBDT, and random forest.

In the present case, we use SVM, GBDT, and random forest. The SVM model can implicitly map inputs into high-dimensional feature spaces and perform non-linear classification using what is called the kernel trick. The GBDT model can distinguish a variety of features and combine these features. The random forest model is capable of handling high-dimensional features without having to select features during fast training.

3.3.1 SVM model building

SVM[17] is a popular methodology for binary classification and a number of modified formulations have been derived from it. Consider a set of training vectors $\{x_i \in \mathbf{R}^p,\ i = 1,\ldots,m\}$ and its corresponding set of labels $\{y_i \in \{-1,1\},\ i = 1,\ldots,n\}$, this model can predict the class labels of unseen data.

The soft-margin SVM training problem can be expressed as follows:

$$\min_{w,b,\xi} \frac{1}{2}\|w\|^2 + c\sum\nolimits_{i=1}^{n} \xi_i \tag{7}$$

subject to

$$\begin{cases} y_i(w^{\mathrm{T}} + b) + \xi_i, & i = 1,\ldots,n, \\ \xi_i \geqslant 0, & i = 1,\ldots,n \end{cases} \tag{8}$$

where ξ_i is a slack variable associated with a penalty term in the objective with a magnitude controlled by c, a problem specific parameter. Vector w is the vector normal to the separating hyperplane $(w^{\mathrm{T}}x_i + b = 0)$, and b is its position relative to the origin. Formula (7) maximizes the margin $2/\|w\|$ between the two separating hyperplanes $(w^{\mathrm{T}}x_i + b = 1)$ and $(w^{\mathrm{T}}x_i + b = -1)$. The use of slack variable ξ_i penalizes data points that fall on the wrong side of these hyperplanes.

Table 5 Features based on attribute description.

Attribute	Description	Dimension	Value type
cuser-EqualQuser	Same(Not) User of Question and Comment	1	Bool
qCategory-Probility	Question corresponds to all Comment's CGOLD	6	List

We use the NLTK[11] to carry out various natural language processes on the raw documents. The documents are tokenized into raw words, and these words are converted based on morphology and part-of-speech tagging. They are also filtered by stop words provided by the NLTK. Thereafter, we compute the *tf-idf* value of each word in the documents and construct the vector $d_i = (t_{i1},\ldots,t_{im})$, where t_{ik} is the *tf-idf* value of word k in document i of collection D, m is the number of all words used in the collection.

Document similarity is computed by the cosine similarity of vectors using *tf-idf* as weights.

$$sim_{ij}^{vsm} = \frac{\sum_{k=1}^{m} t_{ik}t_{jk}}{\sqrt{\sum_{k=1}^{m} t_{ik}^2}\sqrt{\sum_{k=1}^{m} t_{jk}^2}},\ t_{ik} = tf_{ik}\cdot idf_{kD} \tag{9}$$

3.3.2 GBDT model building

GBDT[18] is a gradient boosting algorithm that utilizes decision stumps or regression tress as weak classifiers. In GBDT, weak learners measure the error observed in each node, split the node using a test function $\kappa : \mathbf{R}^n \to \mathbf{R}$ with a threshold τ , and then return the values η^l and η^r. The optimal split of (τ, η^l, η^r) to minimize the error after the split is given by

$$\varepsilon(\tau) = \sum_{i:\kappa(x_i)<\tau} w_i^j (r_i^j - \eta^l)^2 + \sum_{i:\kappa(x_i)\geqslant\tau} w_i^j (r_i^j - \eta^r)^2 \tag{10}$$

where w_i^j and r_i^j respectively denote the weight and response of x_i in the j-th iteration. Formally, these terms are expressed as

$$w_i^j = \exp(-y_i f_{j-1}(x_i)) \tag{11}$$

$$r_i^j = g(x_i)/w_i^j = -y_i\exp(-y_i f_{j-1}(x_i))/w_i^j = -y_i \tag{12}$$

We identify the optimal triplet $(\tau^*, \eta^{l*}, \eta^{r*})$ by minimizing the error over all possible τ's at each node, given τ and (η^{l*}, η^{r*}) can be found simply by computing the weighted average of r_i^j's over training examples that fall on the corresponding side of the split. η^{l*} and η^{r*} are given to the left and right children of the current node, respectively, and η's stored in the leaf node are used as a scores of weak learners corresponding to the tree depending on its input x.

Reference [19] provides more details on this algorithm.

3.3.3 Random forest model building

The random forest algorithm[20] is an ensemble classifier algorithm based on the decision tree model. It generates k different training data subsets from an original dataset using a bootstrap sampling approach. Each sample of the testing dataset is predicted by all decision trees, and the final classification result is returned depending on the votes of these trees. Random forest models perform well in classification tasks and work efficiently on large datasets.

We train and save a random forest classification model based on the features we extracted. We apply the random forest model to the continuous input of the training set and collect false positives and negatives, which are examples of intervals from the training set that the classifier fails to classify correctly. The sets of false positive and negative instances are then added to the original training set, and the random forest is retrained using the new extended set of training examples.

4 Experimental Evaluation

We provide an experimental study of our method for evaluation. This demonstration uses the original CQA-QL corpus[2] shown in Table 6. The dataset consists of 3229 questions. The total number of comments is 20 162, 9941 of which were good (49.31%), 2013 of which were potential (9.98%), and 8208 of which were bad (40.71%) comments; each question had an average of 6.24 comments. In addition, each question had a title and description and answers.

Table 6 Statistics of CQA-QL corpus dataset.

Category	Train	Dev.	Test
Questions	2600	300	329
-GENERAL	2376	266	304
-YES/NO	224	34	25
Comments	16 541	1645	1976
-min per question	1	1	1
-max per question	143	32	66
-avg per question	6.36	5.48	6.01
CGOLD values	16 541	1645	1976
-Good	8069	875	997
-Potential	1659	187	167
-Bad	6813	583	812
CGOLD_YN values	795	115	111
-Yes	346	62	–
-No	236	32	–
-Unsure	213	21	–
QGOLD_YN values	224	34	25
-Yes	87	16	15
-No	47	8	4
-Unsure	90	10	6

Compared with those of the training data set, the *CGOLD* value of the label distribution was basically similar but the *QGOLD_YN* value of the label distribution was relatively large. Our experiments were performed on a cluster of four machines running on Ubuntu 14.04 LTS. Each node was equipped with an Intel Xeon E5-2609 2.5 GHz CPU and 32 GB of memory.

We used *train*, *devel*, and *test* datasets to train the LSI, LDA, SenLDA, Word2Vec, and BOW models to extract similarity features. The SVM, random forest, and GBDT models were also trained. We performed the experiments by setting the general number class to 6 and the *YES/NO* number class equal to 3. In the SVM model, we set the parameter C to 1 and *gamma* equal to 0.001 in the *rbf* kernel function. In random forest model, the max depth in the *DecisionTreeClassifier* function was set to 4. In the GBDT model, the parameter *n_estimators* was set to 1000, *learning_rate* was set to 0.2, and *max_depth* was set to 2 in the function *GradientBoostingClassifier*.

Experiments were performed by Macro F1, Accuracy, and Ranking according to the *test* dataset. The performance of differentiating *good* and *bad* answers are among all answers.

We compared our method against the JAIST[21], HITSZ-ICRC[22], QCRI[23], ECNU[24], ICRC-HIT[25], VectorSlu[26], Shiraz[27], FBK-HLT[28], Voltron[29], and CICBUAPnlp[30].

5 Results and Discussion

Table 7 shows the results of Task A. In the evaluation, only the measurement of the *Good*, *Potential*, and *Bad* three labels are conducted.

JAIST features are extracted from the answers (with their questions treated as the context), and the feature values are passed through a regression model. In our proposed method, we apply SVM, GBDT, and random forest classifiers to select high-quality result with a ranking function, similar to HITSZ-ICRC. However, the latter uses syntax and deep semantic features to improve its performance. The features determined by QCRI use a supervised machine learning approach and a manifold of features, including word n-grams,

Table 7 Result of Task A.

Model	Macro F1 (%)	Accuracy (%)	Ranking
Baseline	22.36	50.46	–
JAIST	57.19	72.52	1
HITSZ-ICRC	56.41	68.67	2
QCRI	53.74	70.50	3
ECNU	53.47	70.55	4
ICRC-HIT	49.60	67.68	5
VectorSlu	49.10	66.45	6
Shiraz	47.34	56.83	7
FBK-HLT	47.32	69.13	8
GBDT	**46.9**	**68.12**	**9**
Random Forest	**46.74**	**65.89**	**10**
SVM	**43.35**	**43.35**	**11**
Voltron	46.07	62.35	12
CICBUAPnlp	40.40	53.74	13

Table 8 Result of Task B.

Model	Macro F1 (%)	Accuracy (%)	Ranking
Baseline	25.0	60	–
VectorSlu	63.7	72.0	1
GBDT	**59.14**	**65.59**	**2**
ECNU	55.8	68.0	3
QCRI	53.6	64.0	4
HITSZ-ICRC	53.6	64.0	5
Random Forest	**52.86**	**65.52**	**6**
SVM	**47.34**	**58.63**	**7**
CICBUAPnlp	38.8	44.0	8
ICRC-HIT	30.9	52.0	9
FBK-HLT	27.8	40.0	10

text similarity, sentiment dictionaries, the presence of specific words, the context of a comment, and some heuristics. CICBUAPnlp only uses syntactic and morphological features to compare the structures of answers with the structures of labeled sets.

Our methodology is quite straightforward in its combination of selected features and use of specific models to train the data. We then conduct differential selection from the features and models. In brief, all the evaluation results are better than the baseline by the indicators Macro F1 and Accuracy and achieve the good performance. Although, our results did not overcome the general average, especially such as JAIST[21], HITSZ-ICRC[22], and QCRI[23] (top three). Evaluation on the GBDT model is clearly better compared with that on the SVM and random forest models.

Table 8 summarizes the results of our submitted runs on Task B datasets officially released by the organizers; the top rank runs are also provided. Each question is given Yes, No, and Unsure labels. The reported results also include the results of the baseline model and the best result. All of the evaluation results demonstrate the good performance of the proposed method.

In this experiment, the results of the GBDT and random forest models surpass the general average (47.54%). The rank of the GBDT model is 2. These results indicate that the random forest and SVM models can show reasonable performance in the *Yes* and *Unsure* classes but are unble to obtain the *No* class. Moreover, most of the instances of the *No* class are misclassified to the *Unsure* class.

By analyzing features and model building, we believe the experiment can also be improved in the following areas:

(1) This experiment only uses the traditional machine learning method and does not apply deep learning algorithms, such as Convolutional Neural Networks (CNNs), which may show good performance.

(2) This experiment only randomly selects some of the super-parameters of the model, which are not fine-grained. Ultra-parameter selection options can be analyzed in future work.

(3) Only 20 features were selected in this study; more features could be added in later studies.

Despite these limitations, the experimental results show that our approach is efficient, and its overall rank is fairly high compared with those of other methods.

6 Related Work

Several works have focused on the quality of answers on CQA websites[31–33], which is greatly valuable for information retrieval. Prior studies on answer selection generally treat the task as a classification problem, that relies on exploring various features to represent the QA pair. A number of researchers have attempted to explore and determine various features that define question quality[16, 34–36]. In Ref. [37], for example, the authors designed specific features and applied structure prediction models. The authors of Ref. [38] used various types of features including 8 similarity features, 44 heuristic features, and 16 thread-based features. Although these methods achieved good performance, they also rely heavily on feature engineering, which requires a large amount of manual work and domain expertise. In Ref. [39], the authors integrated textual with structural features to represent the candidate pairs and then applied SVM to classify the candidate pairs. In Ref. [40], the authors used a GBDT as the classifier, and built ensembles from different boosted decision

tree models to improve prediction accuracy. In Ref. [41], an approach based on conditional random fields was proposed; this approach could capture contextual features for semantic matching between question and answer. Similarly, we identified effective features in content dimentional way. Furthermore, we applied SVM, GBDT, and random forest classifiers to select high-quality results with a ranking function.

Some studies have exploited syntactic tree structures to measure the semantic correspondence between question and answer, such as Refs. [42, 43]. In Refs. [44, 45], the authors proposed directly learning the distributed representation of QA pairs. The work in Ref. [46] demonstrated that 2D convolutional sentence models can represent the hierarchical structures of sentences and capture rich matching patterns between two language objects. In Ref. [47], the authors formulated answer selection as a semantic matching problem with a latent word-alignment structure and conducted a series of experimental studies on leveraging proposed lexical semantic models. One disadvantage of these approaches is that semantic correlations among answers and questions, which are very important for answer selection, are ignored.

The translation-based language model has also been used for QA matching by transferring the answer to the corresponding question[48, 49]. However, this method suffers from informal words or phrases in Q&A archives and shows limited applicability in new domains. In Ref. [49], a retrieval model that combines a language model for the question part with a query likelihood approach for the answer part was proposed.

Recently, deep learning-based approaches have recently been applied. In Ref. [44], for example, the authors proposed a deep belief net-based semantic relevance model to learn the distributed representation of QA pairs. The CNN-based sentence representation models have achieved successes in neural language processing tasks. In Ref. [50], the authors applied CNNs to learn the joint representation of QA pairs and then used the joint representation as inputs of the Long Short-Term Memory algorithm to learn the answer sequence of a question and label the matching quality of each answer. In Ref. [51], the authors trained a logistic regression classifier with user metadata to predict the quality of answers; this approach may show good performance, and we will consider employing it in the next phase of our research.

7 Conclusion

Answer selection in CQA is a challenging task in natural language processing. In this paper, we enhanced answer selection in CQA using multi-dimensional feature combination including two subtasks: (a) classifying answers as *good*, *bad*, or *potentially relevant* with respect to the question and (b) answering a *YES/NO* question based on the list of all answers. Two methods are proposed: using traditional QA based on the similarity sorting method and using the information of the answer to the question to find similarities between question and answer. We then use the answer text description information to judge whether the answer is reasonable. We first extract the attribute and content information from each question and comment from the data set. Then, we build models from these features and use SVM, GBDT, and random forest to classify them. The experimental results show that our approach is significantly more efficient than the baseline model, and its overall ranking is relatively high compared with those of other methods.

In the feature, we will use other models, such as neural networks, for experiments on model building and logistic regression to train the dataset. In addition, strong dependencies were observed among the different answers to the same problem.

Acknowledgment

This research was developed by the NLP601 group at School of Electronics Engineering and Computer Science, Peking University, within the National Natural Science Foundation of China (No. 61672046).

References

[1] M. Asaduzzaman, A. S. Mashiyat, C. K. Roy, and K. A. Schneider, Answering questions about unanswered questions of stack overflow, in *Proceedings of the 10th Working Conference on Mining Software Repositories*, San Francisco, CA, USA, 2013, pp. 97–100.

[2] P. Nakov, L. Mrquez, A. Moschitti, W. Magdy, H. Mubarak, A. A. Freihat, J. Glass, and B. Randeree, SemEval-2016 task 3: Community question answering, in *Proceedings of the 10th International Workshop on Semantic Evaluation*, San Diego, CA, USA, 2016, pp. 525–545.

[3] A. Baltadzhieva and G. Chrupala, Question quality in community question answering forums: A survey, *SIGKDD Explorations*, vol. 17, no. 1, pp. 8–13, 2015.

[4] E. Agichtein, C. Castillo, D. Donato, A. Gionis, and G. Mishne, Finding high-quality content in social media, in *Proceedings of the International Conference on Web Search and Web Data Mining*, Palo Alto, CA, USA, 2008, pp. 183–194.

[5] B. Li, T. Jin, M. R. Lyu, I. King, and B. Mak, Analyzing and predicting question quality in community question answering services, in *Proceedings of the 21st World Wide Web Conference*, Lyon, France, 2012, pp. 775–782.

[6] S. C. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, and R. A. Harshman, Indexing by latent semantic analysis, *Journal of the American Society for Information Science*, vol. 41, no. 6, pp. 391–407, 1990.

[7] D. M. Blei, A. Y. Ng, and M. I. Jordan, Latent Dirichlet allocation, *Journal of Machine Learning Research*, vol. 3, pp. 993–1022, 2003.

[8] G. Balikas, M. Amini, and M. Clausel, On a topic model for sentences, in *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*, Pisa, Italy, 2016, pp. 921–924.

[9] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, Distributed representations of words and phrases and their compositionality, in *Proceedings Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems*, Lake Tahoe, NV, USA, 2013, pp. 3111–3119.

[10] T. Mikolov, K. Chen, G. Corrado, and J. Dean, Efficient estimation of word representations in vector space, in *Proceedings of Workshop ICLR*, 2013.

[11] S. Bird, E. Klein, and E. Loper, *Natural Language Processing with Python*. Dublin, Ireland: O'Reilly Press, 2009.

[12] C. D. Manning, P. Raghavan, and H. Schtze, *Introduction to Information Retrieval*. Cambridge, England: Cambridge University Press, 2008.

[13] G. Heinrich, Parameter estimation for text analysis, Technical report, Fraubhofer IGD, 2005, pp. 1–31.

[14] X. Chen, L. Xu, Z. Liu, M. Sun, and H. Luan, Joint learning of character and word embeddings, in *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence*, Buenos Aires, Argentina, 2015, pp. 1236–1242.

[15] A. Abbasi, H. Chen, and A. Salem, Sentiment analysis in multiple languages: Feature selection for opinion classification in web forums, *ACM Trans. Inf. Syst*., vol. 26, no. 3, pp. 12:1–12:34, 2008.

[16] J. Liang, X. Zhou, L. Guo, and S. Bai, Feature selection for sentiment classification using matrix factorization, in *Proceedings of the 24th International Conference on World Wide Web Companion*, Florence, Italy, 2015, pp. 63–64.

[17] G. Salton, A. Wong, and C. Yang, A vector space model for automatic indexing, *Commun. ACM*, vol. 18, no. 11, pp. 613–620, 1975.

[18] J. Xie and S. Coggeshall, Prediction of transfers to tertiary care and hospital mortality: A gradient boosting decision tree approach, *Statistical Analysis and Data Mining*, vol. 3, no. 4, pp. 253–258, 2010.

[19] V. L. C. Becker, R. Rigamonti, and P. Fua, Supervised feature learning for curvilinear structure segmentation, in *Proceedings of Medical Image Computing and Computer-Assisted Intervention*, Nagoya, Japan, 2013, pp. 526–533.

[20] L. Breiman, Random forests, *Machine Learning*, vol. 45, no.1, pp. 5–32, 2001.

[21] Q. H. Tran, V. Tran, T. Vu, M. Ng, and S. B. Pham, JAIST: Combining multiple features for answer selection in community question answering, in *Proceedings of the 9th International Workshop on Semantic Evaluation*, Denver, CO, USA, 2015, pp. 215–219.

[22] Y. Hou, C. Tan, X. Wang, Y. Zhang, J. Xu, and Q. Chen, HITSZ-ICRC: Exploiting classification approach for answer selection in community question answering, in *Proceedings of the 9th International Workshop on Semantic Evaluation*, Denver, CO, USA, 2015, pp. 196–202.

[23] M. Nicosia, S. Filice, A. B. Cedeo, I. Saleh, H. Mubarak, W. Gao, P. Nakov, G. Martino, A. Moschitti, K. Darwish, et al., QCRI: Answer selection for community question answering-experiments for Arabic and English, in *Proceedings of the 9th International Workshop on Semantic Evaluation*, Denver, CO, USA, 2015, pp. 203–209.

[24] J. Zhao, T. Zhu, and M. Lan, ECNU: One stone two birds—Ensemble of heterogenous measures for semantic relatedness and textual entailment, in *Proceedings of the 8th International Workshop on Semantic Evaluation*, Dublin, Ireland, 2014, pp. 271–277.

[25] X. Zhou, B. Hu, J. Lin, Y. Xiang, and X.Wang, ICRC-HIT: A deep learning based comment sequence labeling system for answer selection challenge, in *Proceedings of the 9th International Workshop on Semantic Evaluation*, Denver, CO, USA, 2015, pp. 210–214.

[26] Y. Belinkov, M. Mohtarami, S. Cyphers, and J. R. Glass, VectorSLU: A continuous word vector approach to answer selection in community question answering systems, in *Proceedings of the 9th International Workshop on Semantic Evaluation*, Denver, CO, USA, 2015, pp. 282–287.

[27] A. H. Alashty, S. Rahmani, M. Roostaee, and M. Fakhrahmad, A proposed list wise approach to answer validation, in *Proceedings of the 9th International Workshop on Semantic Evaluation*, Denver, CO, USA, 2015, pp. 220–225.

[28] N. Phuoc, S. Magnolini, and O. Popescu, FBK-HLT: An application of semantic textual similarity for answer selection in community question answering, in *Proceedings of the 9th International Workshop on Semantic Evaluation*, Denver, CO, USA, 2015, pp. 231–235.

[29] I. Zamanov, M. Kraeva, N. Hateva, I. Yovcheva, I. Nikolova, and G. Angelova, Voltron: A hybrid system for answer validation based on lexical and distance features, in *Proceedings of the 9th International Workshop on Semantic Evaluation*, Denver, CO, USA, 2015, pp. 242–246.

[30] H. G. Adorno, D. Vilario, D. Pinto, and G. Sidorov, CICBUAPnlp: Graph-based approach for answer selection in community question answering task, in *Proceedings of the 9th International Workshop on Semantic Evaluation*, Denver, CO, USA, 2015, pp. 18–22.

[31] J. Lou, K. H. Lim, Y. Fang, and J. Z. Peng, Drivers of knowledge contribution quality and quantity in online question and answering communities, in *Proceedings of Pacific Asia Conference on Information Systems*, Queensland, Australia, 2011, p. 121.

[32] J. Jeon, W. B. Croft, J. H. Lee, and S. Park, A framework to predict the quality of answers with non-textual features, in *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Washington, DC, USA, 2006, pp. 228–235.

[33] M. A. Suryanto, E. P. Lim, A. Sun, and R. H. L. Chiang, Quality-aware collaborative question answering: Methods and evaluation, in *Proceedings of the Second International Conference on Web Search and Web Data Mining*, Barcelona, Spain, 2009, pp. 142–151.

[34] L. Mamykina, B. Manoim, M. Mittal, G. Hrip, and B. Hartmann, Design lessons from the fastest q&a site in the west, in *Proceedings of the International Conference on Human Factors in Computing Systems*, 2011, pp. 2857–2866.

[35] B. Wang and L. Sun, Extracting Chinese question answer pairs from online forums, in *Proceedings of the IEEE International Conference on Systems*, 2009, pp. 1159–1164.

[36] C. Shah and J. Po, Evaluating and predicting answer quality in community qa, in *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, Geneva, Switzerland, 2010, pp. 411–418.

[37] A. B. Cedeo, S. Filice, G. D. Martino, S. R. Joty, L. Mrquez, P. Nakov, and A. Moschitti, Thread-level information for comment classification in community question answering, in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing*, Beijing, China, 2015, pp. 687–693.

[38] S. Filice, D. Croce, A. Moschitti, and R. Basili, KeLP at SemEval-2016 task 3: Learning semantic relations between questions and answers, in *Proceedings of the 10th International Workshop on Semantic Evaluation*, 2016, pp. 1116–1123.

[39] J. Huang, M. Zhou, and D. Yang, Extracting chatbot knowledge from online discussion forums, in *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, Hyderabad, India, 2007, pp. 423–428.

[40] H. Lu and M. Kong, Community-based question answering via contextual ranking metric network learning, in *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, 2017, pp. 4963–4964.

[41] S. Ding, G. Cong, C. Lin, and X. Zhu, Using conditional random fields to extract contexts and answers of questions from online forums, in *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics*, 2008, pp. 710–718.

[42] K. Wang, Z. Ming, and T. Chua, A syntactic tree matching approach to finding similar questions in community based qa services, in *Proceedings of the 32nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2009, pp. 187–194.

[43] A. Moschitti, S. Quarteroni, R. Basili, and S. Manandhar, Exploiting syntactic and shallow semantic kernels for question answer classification, in *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*, Prague, Czech Republic, 2007, pp. 776–783.

[44] B. Wang, X. Wang, C. Sun, B. Liu, and L. Sun, Modeling semantic relevance for question-answer pairs in web social communities, in *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, Uppsala, Sweden, 2010, pp. 1230–1238.

[45] H. Hu, B. Liu, B. Wang, M. Liu, and X. Wang, Multimodal dbn for predicting high-quality answers in cqa portals, in *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, Sofia, Bulgaria, 2013, pp. 843–847.

[46] B. Hu, Z. D. Lu, H. Li, and Q. Chen, Convolutional neural network architectures for matching natural language sentences, in *Proceedings of Annual Conference on Neural Information Processing Systems*, Quebec, Canada, 2014, pp. 2042–2050.

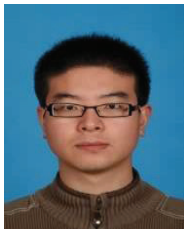
[47] W. Yih, M. W. Chang, C. Meek, and A. Pastusiak, Question answering using enhanced lexical semantic models, in *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, Sofia, Bulgaria, 2013, pp. 1744–1753.

[48] J. Jeon, W. B. Croft, and J. Ho Lee, Finding similar questions in large question and answer archives, in *Proceedings of the 2005 ACM CIKM International Conference on Information and Knowledge Management*, Bremen, Germany, 2005, pp. 84–90.

[49] X. Xue, J. Jeon, and W. B. Croft, Retrieval models for question and answer archives, in *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Singapore, 2008, pp. 475–482.

[50] X. Zhou, B. Hu, Q. Chen, B. Tang, and X. Wang, Answer sequence learning with neural networks for answer selection in community question answering, in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics*, Beijing, China, 2015, pp. 713–718.

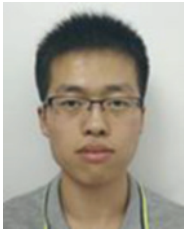
[51] C. Shah and J. Pomerantz, Evaluating and predicting answer quality in community QA, in *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, Geneva, Switzerland, 2010, pp. 411–418.



Hongjie Fan received the master degree from Peking University in 2010. He is working toward the PhD degree at Peking University. His research interests include data exchange technology, schema matching, etc.

Zhiyi Ma is an associate professor at Peking University. His research interests include software modeling technology and model driven development. He received the PhD degree from Northeastern University, China, in 1999.

Hongqiang Li received the bachelor degree from China University of Geosciences, Beijing, in 2015. He is working toward the master degree in software engineering at Peking University. His research interests include information retrieval, machine learning, etc.



Dongsheng Wang received the bachelor degree from Northeastern University, China, in 2016. He is working toward the master degree in software engineering at Peking University. His research interests include data exchange technology, information retrieval, etc.

Junfei Liu is a professor and PhD supervisor at School of Electronics Engineering and Computer Science, Peking University. His research interests include software engineering, information exchange technology, and smart city. He received the BS degree in 1985 from Hunan University, China, and completed the MS degree in National Defense University, China, in 1988. Then, he earned PhD degree from Peking University, China, in 1994.