Data-Driven Answer Selection in Community QA Systems

Liqiang Nie, Xiaochi Wei, Dongxiang Zhang, Xiang Wang, Zhipeng Gao, and Yi Yang

Abstract—Finding similar questions from historical archives has been applied to question answering, with well theoretical underpinnings and great practical success. Nevertheless, each question in the returned candidate pool often associates with multiple answers, and hence users have to painstakingly browse a lot before finding the correct one. To alleviate such problem, we present a novel scheme to rank answer candidates via pairwise comparisons. In particular, it consists of one offline learning component and one online search component. In the offline learning component, we first automatically establish the positive, negative, and neutral training samples in terms of preference pairs guided by our data-driven observations. We then present a novel model to jointly incorporate these three types of training samples. The closed-form solution of this model is derived. In the online search component, we first collect a pool of answer candidates for the given question via finding its similar questions. We then sort the answer candidates by leveraging the offline trained model to judge the preference orders. Extensive experiments on the real-world vertical and general community-based question answering datasets have comparatively demonstrated its robustness and promising performance. Also, we have released the codes and data to facilitate other researchers.

Index Terms—Community-based question answering, answer selection, observation-guided training set construction

1 Introduction

Community question answering system (cQA), one of the fastest-growing user-generated-content (UGC) portals, has risen as an enormous market, so to speak, for the fulfillment of complex information needs. cQA enables users to ask/answer questions and search through the archived historical question-answer (QA) pairs. Compared to the traditional factual QA, such as "who is the president of the Singapore in 2016", which can be answered by simply extracting named entities or paragraphs from documents, cQA have made substantial headway in answering complex questions, such as reasoning, open-ended, and advice-seeking questions. cQA is thus quite open and has little restrictions, if any, on who can post and who can answer a question. The past decade has witnessed the significant society value of both the general cQA sites, such as Yahoo! Answers[1] and Quora,[2] and the vertical ones like Stack Overflow[3] and HealthTap.[4]

1. https://answers.yahoo.com/
2. https://www.quora.com/
3. http://stackoverflow.com/
4. https://www.healthtap.com/

- *L. Nie is with the School of Computer Science and Technology, Shandong University, Jinan Shi 250000, China. E-mail: nieliqiang@gmail.com.*
- *X. Wei, Z. Gao, and X. Wang are with the School of Computing, National University of Singapore, Singapore 119077. E-mail: xcwei.bit@gmail.com, xiangwang@u.nus.edu, beyond_acm@163.com.*
- *D. Zhang is with the School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu Shi 610051, China. E-mail: zhangdongxiang37@gmail.com.*
- *Y. Yang is with the University of Technology Sydney, Ultimo, NSW 2007, Australia. E-mail: yee.i.yang@gmail.com.*

Manuscript received 8 June 2016; revised 17 Jan. 2017; accepted 4 Feb. 2017. Date of publication 15 Feb. 2017; date of current version 27 Apr. 2017. Recommended for acceptance by H. Lee.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TKDE.2017.2669982

Despite the success of cQA and active user participation, question starvation widely exists in cQA forums, which refers to the following two kinds of phenomena:

- First, information seekers usually have to wait a long time before getting answers to their questions. For instance, a study [1] over 200 thousand questions in Yahoo! Answers reported that it takes on average more than half an hour to receive the first answers if the questions are raised in the evening, and the time is more than double if the questions are posted in the morning. Comparatively speaking, the waiting time is much longer in the vertical cQA, such as Healthtap [2], spanning from hours to days.
- Second, a large proportion of questions do not get any response even within a relatively long period. Considering Yahoo! Answers as an example, around 15 percent of its questions do not receive any answer and leave the askers unsatisfied [3]. Even worse is the Wikianswers.[5] As reported on its official website upon approximately one million questions, only 27 percent of them are answered.

Question starvation is probably caused by several reasons: 1) the questions are poorly phrased, ambiguous or not interesting at all; 2) the cQA systems are hardly to route the newly posted questions to the appropriate answerers; and 3) the potential answerers have the matched expertise, but are not available or overwhelmed by the sheer volume of incoming questions. This case often occurs in the vertical cQA forums, whereby only authorized experts are allowed to answer these questions. Regarding the first case, question quality modeling has been well-studied [4], [5], which can

5. http://answers.wikia.com/wiki/Wikianswers

1041-4347 © 2017 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

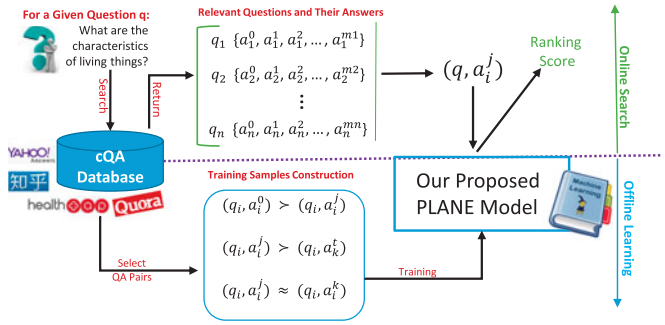


Fig. 1. Illustration of our proposed answer ranking scheme. Thereinto, a_i^j denotes the jth answer of the ith question q_i and a_i^0 refers to the best answer of q_i.

assess the question quality and serve to remind askers to rephrase their questions. For the latter two cases, great efforts have been dedicated to lightening their situations via the so-called question routing [6], [7], by considering the expertise matching [8] and availability of potential answerers [9].

Question routing works by exploring the current system resources, especially the human resources. Beyond that, we can reuse the past solved questions to answer the newly asked ones. Indeed, a tremendous number of historical QA pairs, as time goes on, have been archived in the cQA databases. Information seekers hence have large chances to directly get the answers by searching from the repositories, rather than the time-consuming waiting. Inspired by this, Wang et al. [10] have transformed the task of QA to the task of finding relevant and similar questions. However, the returned top question candidates usually associate with multiple answers,[6] and the research on choosing the right answers from the relevant question pool is relatively sparse.

Given a question, instead of naively choosing the best answer from the most relevant question, in this paper, we present a novel Pairwise Learning to rANk model, nicknamed PLANE, which can quantitatively rank answer candidates from the relevant question pool. Fig. 1 demonstrates the workflow of the PLANE model, consisting of two components: offline learning and online search. Particularly, during the offline learning, guided by our user studies and observations, we automatically establish the positive, negative, and neutral training samples in terms of preference pairs. The PLANE model can be jointly trained with these three kinds of training samples. As a byproduct, it is able to identify the discriminative features by a ℓ_1 regularizer. To optimize the PLANE model, we approximate it with a quadratically smoothed hinge function and a smooth convex approximation of lasso. Therewith the approximation, we derive its closed-form solution. When it comes to the online search, for a given question, we pair it with each of the answer candidates, and fit them into the trained PLANE model to estimate their matching scores. To verify our proposed model, we conduct extensive experiments over two datasets, collected from a vertical cQA site HealthTap and a general cQA site Zhihu.com, respectively. For each QA pair, we extract a comprehensive set of features for the descriptive representation. By comparing with several state-of-the-art baselines, the superiority of our proposed PLANE model is demonstrated.

6. According to our study on over 114,200 solved questions in Zhihu.com, each question has 5.07 answers on average.

In summary, we have three main contributions:

- Inspired by our user studies and observations, we present a novel approach to constructing the positive, neutral, and negative training samples in terms of preference pairs. This greatly saves the time-consuming and labor-intensive labeling process.
- We propose a pairwise learning to rank model for answer selection in cQA systems. It seamlessly integrates hinge loss, regularization, and an additive term within a unified framework. Different from the traditional pairwise learning to rank models, ours incorporates the neutral training samples and learns the discriminative features. In addition, we have derived its closed-form solution by equivalently reformulating the objective function into a smoothed and differentiable one.
- We have released the codes and datasets to facilitate other researchers to repeat our work and verify their ideas.[7]

The remainder is structured as follows. Section 2 reviews the related work. Sections 3 and 4 introduce the offline learning and online search, respectively. Experimental settings and results are reported in Section 5, followed by the conclusion and future work in Section 6.

2 Related Work

Four threads of literature are relevant to our work.

2.1 Feature-Driven Answer Selection

Conventional techniques for filtering answers primarily focus on generating complementary features relying on the highly structured cQA sites. Jeon et al. [5] extracted a set of non-textual features covering the contextual information of QA pairs, and proposed a language model for processing these features in order to predict the quality of answers collected from a specific cQA service. Two years later, Liu et al. [11] found powerful features including structural, textual, and community features, and leveraged the traditional shallow learning methods to combine these heterogeneous features. Blooma et al. [12] developed a hierarchical framework to identify the predictive factors for obtaining a high-quality answer based on textual and non-textual features. Beyond textual features, Nie et al. [13] explored a set of features extracted from media entities, such as color, shape and bag-of-visual-words. Following them, Ding et al. [1] introduced a general classification framework to combine the evidence from different views, including the graph-based relationship, content, and usage-based features. In recent years, the authors in [14] described their system for SemEval-2015 Task 3: Answer Selection in cQA. The system combines 16 features from five groups to predict answer quality, and achieves the best performance in subtask A for English, both in accuracy and F1-score. Most recently, Wei et al. [15] extracted heterogeneous features from three-level cQA structures, i.e., category-level, question-level, and answer-level. They successfully enhance the performance of answer ranking with these features.

7. The codes and data can be publicly accessible via: http://datapublication.wix.com/tkde-plane

2.2 Learning to Rank

Density-based ranking [16] is one of the early methods for answer selection, which considers the surface distance between question terms and the answer target. It, however, may fare poorly with questions whose answer target type is unknown and is ineffective in handling lexical level matching. To overcome such problem, researchers [17], [18] turned to leverage dependency relations between matched question terms and the answer target as additional evidence to select the correct answer, and achieved promising performance. Beyond matching, Surdeanu et al. [19] and Agarwal et al. [20] respectively combined a few types of machine learning methodologies into a single model and validated its capability of dealing with factoid and complex questions, using a large number of, possible noisy, QA pairs. Apart from the aforementioned shallow learning methods, a team from Emory University [21] trained a Long Short-Term Memory neutral network model to automatically and promptly reply questions in cQA. Theoretical research serves practice. For example, Anna Shtok et al. [3] attempted to reduce the rate of unanswered questions in Yahoo! Answers by reusing the large repository of past resolved questions, openly available on the site. In particular, they presented a two-stage QA algorithm that first identifies the past resolved questions relevant to the given one, and then applied simple classifiers to justify whether the corresponding past answers meets the new question needs.

The answer retrieval problem in cQA is similar to the traditional ranking task [22], [23], whereby the given question and the set of answer candidates are analogous to a query and a set of relevant entities such as images and documents, respectively [20]. The target is now transformed to find an optimal ranking order of these answer candidates according to their relevance/correctness/quality to the given query. Learning a ranking function with binary relevance judgments can be achieved in three ways: 1) Pointwise. Methods in this way [24], [25] estimate the relevance score of each QA pair individually by a standard classification or a regression model. 2) Pairwise. These methods [26], [27], [28], [29] work by predicting the preference of two answer candidates via a binary classifier. And 3) listwise. The complete ranking of all candidate answers to the same question is optimized [30].

Models in this categories are either supervised or semi-supervised models, which require label-intensive annotations to construct training samples, whereas our proposed model is data-driven and it hence does not need the human efforts.

2.3 Answer Ranking via Finding Experts

Rather than directly ranking community answers, some researchers resort to identify users' authority via graph-based link analysis. The techniques of graph-based link analysis have been well-studied in the social network analysis and achieved great success [31], [32], [33]. In the QA task, they assumed that the authoritative users tend to generate high-quality answers [34]. For example, Jurczyk et al. [35] proposed an adaptation of the HITS algorithm to discover experts in QA portals, and demonstrated its effectiveness for discovering authorities by a large-scale empirical evaluation. The HITS-style algorithms establish two types of graph nodes: 1) hubs which group edges to authoritative nodes; and 2) authorities which are sources of information on a given topic. Askers and answerers are respectively regarded as "hubs" and "authorities". Another representative work on graph-based link analysis was developed by Zhang et al. [36]. They leveraged the PageRank-like algorithms to identify users with high expertise. They found that the expertise network is highly correlated to answer quality. This algorithm not only considers how many other people one helped, but also whom he/she helped. The intuition behind is that if B is able to answer A's question, and C is able to answer B's question, C's expertise rank should be boosted not just because C is able to answer a question, but because C can answer a question of B who had some expertise. In a sense, it propagates expertise scores through the QA network.

Approaches in this stream rank answers indirectly by measuring the expertise of the potential answerers. However, there is no guarantee that professional experts always provide high-quality answers. Instead of exploring the answerers, our proposed approach directly treats the QA pair as a document and compares the preference relationships between QA pairs. In other words, we not only consider the relevance between a question and an answer, but also investigate the relevance preference among QA pairs.

TABLE 1
Data Statistics Collected from HealthTap and Zhihu.com, Respectively

cQA Site	Question #	Answer #	Vote #
HealthTap	39,998	58,091	54,833
Zhihu.com	114,200	578,874	1,257,759

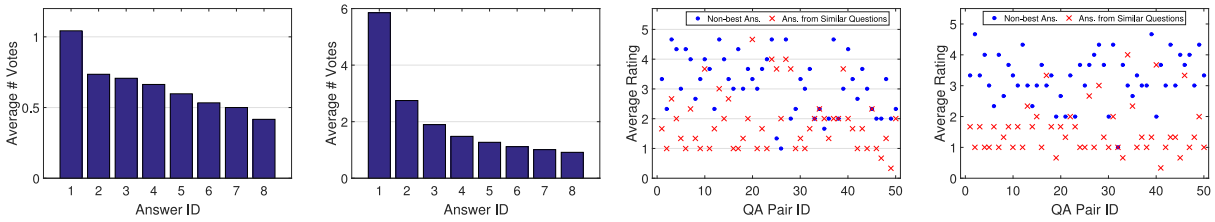
2.4 Preference Pair Construction

Several learning to rank functions are based on the pairwise preference framework, in which instead of taking documents in isolation, document pairs are used as instances in the learning process [37]. The issue of the absolute relevance judgments are the reliability and variability. One possibility to alleviate this problem is to make use of the vast amount of data recording user interactions with the search results, in particular, user clickthroughs data. Each individual user click may not be very reliable, but the aggregation of a great number of user clicks can provide a very powerful indicator of relevance preference [38], [39]. In this regards, Joachims et al. presented approaches to mining logfiles of WWW search engines with the goal of improving their retrieval performance automatically. The key insight is that such clickthrough data can provide training data in the form of relative preferences [40], [41].

3 Offline Learning

3.1 Observation-Guided Training Set Building

To gain the insights into the answer quality in cQA, we collected a set of questions and their answers from the vertical cQA site HealthTap and the general one zhihu.com, respectively. Table 1 summarizes the statistics over these two datasets. For each question, we sorted its answers in decreasing order regarding the number of "votes". Hereafter, we counted the average number of votes over all the answers



(a) Votes on HealthTap (b) Votes on Zhihu.com (c) User Study on HealthTap (d) User Study on Zhihu.com

Fig. 2. Subfigures (a) and (b) illustrate the number of vote distributions over HealthTap and Zhihu.com, respectively. The answers of each question were sorted decreasingly regarding their votes in advance. Subfigures (c) and (d), respectively, display the user study results of QA match over HealthTap and Zhihu.com.

ranked at the same positions. Figs. 2a and 2b illustrate the number of vote distributions over HealthTap and Zhihu.com, respectively. From these two figures, we have the following first two observations:

1) *For a given question, its best answer is preferable to its non-best answers.* In particular, for each question, we found that its best answer is always positioned at the first place in terms of votes. Furthermore, on average, the votes of the best answers far outnumber those of the rest. This reflects that, for a given question, the quality of its best answer is much better than its non-best ones in most cases.
2) *The non-best answers of the same question are almost on a par.* Regarding the non-best answers, we cannot see a significant "vote" drop between two successive ranks, no matter in the vertical or the general cQA sites. This signals that for a given question it is hard to differ the qualities of its non-best answers, from a statistical view.
3) *A question prefers the answers of itself to those of others.* We observed this point from a user study. In particular, we randomly selected 50 questions from our collected HealthTap and Zhihu.com datasets, respectively. For each question, we provided two answers: one was randomly selected from its non-best answers, and the other was randomly selected from those of its similar questions calculated via k-nearest neighbors (k-NN). We invited three volunteers to manually rate the "match" between each QA with a 0-5 grade. The higher grade indicates that the answer is more suitable to the given question. It is worth emphasizing that the volunteers were blinded to which answer is the real one. We averaged the votes and demonstrated the vote distributions in Figs. 2c and 2d. As can be seen, questions' answers, even which are not the best, are always much appreciated by volunteers. This is because, the questions in cQA are often very complex and sophisticated, and hence the question-specific answers are more suitable.

Formally, let us define a_i^j as the jth answer of the ith question q_i, and a_i^0 is the best answer. According to the first and third observations, we have,

$$\begin{cases} (q_i, a_i^0) \succ (q_i, a_i^j), j \neq 0 \\ (q_i, a_i^j) \succ (q_i, a_k^t), i \neq k, \end{cases} \tag{1}$$

where $\succ$ denotes a preference relationship. Let $\mathbf{x} = \mathbf{x}^{(1)} - \mathbf{x}^{(2)}$, where $\mathbf{x}^{(1)}$ and $\mathbf{x}^{(2)}$ respectively denote the D-dimensional feature vectors of the first and second QA pairs in each comparison. Meanwhile, we denote y as the preference relationship of $\mathbf{x}$, which satisfies the conditions below,

$$y = \begin{cases} +1, \text{ if } \mathbf{x}^{(1)} \succ \mathbf{x}^{(2)} \\ -1, \text{ if } \mathbf{x}^{(2)} \succ \mathbf{x}^{(1)}. \end{cases} \tag{2}$$

In the light of this, we can build a training set with preferable labels $\mathcal{X} = \{(\mathbf{x_i}, y_i)\}_{i=1}^N$.

According to the second observation, there does not exist a preference relationship among the non-best answers of a specific question. As a result, it is intuitive to have,

$$(q_i, a_i^j) \cong (q_i, a_i^k), \tag{3}$$

where $\cong$ denotes a neutral preference relationship between the first and second QA pairs. Meanwhile, we constrain $j \neq k$ and $j \times k \neq 0$. To formally formulate the neutral preference, we denote $\mathbf{u} = \mathbf{u}^{(1)} - \mathbf{u}^{(2)}$, where $\mathbf{u}^{(1)}$ and $\mathbf{u}^{(2)}$ refers to the D-dimensional feature vectors of the first and second QA pairs, respectively. Considering all the comparisons in this form, we can create another training set with neutral preference $\mathcal{U} = \{(\mathbf{u_j}, 0)\}_{j=1}^M$.

3.2 Our Proposed PLANE Model

Given a question, we can easily obtain a set of top k relevant questions $\mathcal{Q} = \{q_1, \ldots, q_k\}$ from the archived QA repositories via the well-studied question matching algorithm k-NN. Without loss of generality, we assume question q_i has a set of $m_i \geq 1$ answers, denoted by $\mathcal{A}_i = \{a_i^0, a_i^1, \ldots, a_i^{m_i}\}$, whereby a_i^0 is the best answer of q_i selected by community users. We aim to develop a learning to rank model to sort all the answers associated to the returned relevant questions in $\mathcal{Q}$.

As discussed previously, given a set of QA pairs, we can build the dual training sets $\mathcal{X}$ and $\mathcal{U}$. To jointly incorporate $\mathcal{X}$ and $\mathcal{U}$, we propose the following pairwise learning to rank model,

$$\min_{\mathbf{w}} \sum_{i=1}^{N} \left[1 - y_i \mathbf{w}^T \mathbf{x_i}\right]_+ + \lambda \|\mathbf{w}\|_1 + \mu \sum_{j=1}^{M} |\mathbf{w}^T \mathbf{u}_j|, \tag{4}$$

where $\mathbf{x_i} = \mathbf{x_i}^{(1)} - \mathbf{x_i}^{(2)} \in \mathbb{R}^D$ and $\mathbf{u}_j = \mathbf{u}_j^{(1)} - \mathbf{u}_j^{(2)} \in \mathbb{R}^D$ denotes two training instances from $\mathcal{X}$ and $\mathcal{U}$, respectively; symbols N and M respectively stand for the number of preference pairs in $\mathcal{X}$ and $\mathcal{U}$; and $\mathbf{w} \in \mathbb{R}^D$ represents the desired coefficient vectors.

The first term is a hinge loss function, which is suitable for our binary preference judgment task. It provides a relatively tight and convex upper bound on the 0-1 indicator function. Besides, the empirical risk minimization of this

loss is equivalent to the classical formulation for support vector machine (SVM) [42]. Correctly classified points lying outside the margin boundaries of the support vectors will not be penalized, whereas points within the margin boundaries or on the wrong side of the hyperplane will be penalized in a linear fashion compared to their distance from the correct boundary. The second term is a ℓ_1 norm, which regularizes $\mathbf{w}$ and helps in feature selection. The last term is a sum of absolute values, which aim to penalize the preference distances between non-best answers of the same questions, and it guarantees our second observation in nature.

3.3 Optimization

Although the objective function is convex, it is non-smoothed and not differentiable on $\mathbf{w}$. As we aim to derive its closed-form solution for the global optimal and efficient results, we have to resort the original objective function into a smooth and differentiable one.

3.3.1 Quadratic SVM

Since the derivative of the hinge loss at the condition of $y_i\mathbf{w}^T\mathbf{x_i} = 1$ is non-deterministic, a smoothed version is hence required to ease the optimization. Towards this end, we propose an equivalent quadratic hinge loss as follows,

$$\sum_{i=1}^{N} \left([1 - y_i\mathbf{w}^T\mathbf{x_i}]_+\right)^2, \tag{5}$$

which is quadratically smoothed and differentiable with respect to $\mathbf{w}$.

3.3.2 Reformulation of Lasso

We rewrite the second and third terms of the objective function as follows,

$$\begin{aligned} &\lambda\|\mathbf{w}\|_1 + \mu\sum_{j=1}^{M}|\mathbf{w}^T\mathbf{u}_j| \\ &\equiv \lambda\|\mathbf{I}\mathbf{w}\|_1 + \mu\|\mathbf{U}\mathbf{w}\|_1 \equiv \|\mathbf{C}\mathbf{w}\|_1, \end{aligned} \tag{6}$$

where $\mathbf{I} \in \mathbb{R}^{D\times D}$ is an identity matrix, $\mathbf{U} = (\mathbf{u}_1, \ldots, \mathbf{u}_M)^T \in \mathbb{R}^{M\times D}$, and $\mathbf{C} = \binom{\lambda\mathbf{I}}{\mu\mathbf{U}} \in \mathbb{R}^{(D+M)\times D}$. Notably, $\mathbf{U}$ and $\mathbf{C}$ are constant matrices. Recall that the dual norm of the entry-wise matrix l_∞ norm is the l_1 norm (and vice versa),[8] we rewrite Eqn. (6) as follows,

$$f_0(\mathbf{w}) = \|\mathbf{C}\mathbf{w}\|_1 \equiv \max_{\|A\|_\infty\leq 1} tr(\mathbf{A}^T\mathbf{C}\mathbf{w}), \tag{7}$$

where $\mathbf{A} \in \mathcal{A} = \{\mathbf{A}|\|\mathbf{A}\|_\infty \leq 1, \mathbf{A} \in \mathbb{R}^{(D+M)}\}$. To tackle this non-smooth formulation and make it differentable, we construct its smooth approximation using an auxiliary convex function as follows,

$$f_\nu(\mathbf{w}) = \|\mathbf{C}\mathbf{w}\|_1 \equiv \max_{\|A\|_\infty\leq 1} tr(\mathbf{A}^T\mathbf{C}\mathbf{w}) - \frac{\nu}{2}\|\mathbf{A}\|_F^2, \tag{8}$$

We can see that $f_\nu(\mathbf{w})$ is a smooth lower bound and an approximation of $f_0(\mathbf{w})$. The gap is controlled by a positive smoothness parameter ν.

3.3.3 Alternating Optimization

For notation simplicity, let us denote $l_i(\mathbf{w}) = ([1 - y_i\mathbf{w}^T\mathbf{x_i}]_+)^2$. We then can restate our objective function in Eqn. (4) as follows,

$$\Psi(\mathbf{w}, \mathbf{A}) = \sum_{i=1}^{N} l_i(\mathbf{w}) + f_\nu(\mathbf{w}). \tag{9}$$

We adopt the alternating optimization strategy to optimize our objective function. In particular, we solve one variable while fixing the others in each iteration. We keep this iterative procedure until the objective function converges.

By first fixing $\mathbf{w}$, we take the derivative of Eqn. (9) with respect to $\mathbf{A}$ and set it to zero. We can obtain the optimal solution as follows,

$$\mathbf{A}^* = S\left(\frac{\mathbf{C}\mathbf{w}}{\nu}\right), \tag{10}$$

where $S(\cdot)$ is the shrinkage operator defined as follows: for $z \in \mathbb{R}$, $S(z) = z$, if $-1 < z < 1$; $S(z) = 1$, if $z \geq 1$; and $S(z) = -1$, if $z \leq -1$. When it generalizes to a matrix $\mathbf{B}$, $S(\mathbf{B})$ is defined as applying $S(\cdot)$ to every entry of $\mathbf{B}$.

We then fix $\mathbf{A}$ and differentiate Eqn. (9) with respect to $\mathbf{w}$,

$$\frac{\partial\Psi}{\partial\mathbf{w}} = \sum_{i=1}^{N}\frac{\partial l_i(\mathbf{w})}{\mathbf{w}} + \mathbf{C}^T\mathbf{A}, \tag{11}$$

where $\frac{\partial l_i(\mathbf{w})}{\partial\mathbf{w}}$ equals to,

$$\begin{cases} 0, & \text{if} \quad y_i\mathbf{w}^T\mathbf{x_i} > 1; \\ 2y_i^2\mathbf{x_i}\mathbf{x_i}^T\mathbf{w} - 2y_i\mathbf{x_i}, & \text{if} \quad y_i\mathbf{w}^T\mathbf{x_i} \leq 1. \end{cases} \tag{12}$$

Following that, we set the derivative of $\mathbf{w}$ to zero and obtain the optimal solution as

$$\sum_{i\in\mathcal{I}} y_i^2\mathbf{x_i}\mathbf{x_i}^T\mathbf{w} = \sum_{i\in\mathcal{I}} y_i\mathbf{x_i} - \frac{1}{2}\mathbf{C}^T\mathbf{A}, \tag{13}$$

where $\mathcal{I} = \{i|y_i\mathbf{w}^T\mathbf{x_i} \leq 1\}$ is an indicator set. We can easily prove that $\sum_{i\in\mathcal{I}}\mathbf{x_i}\mathbf{x_i}^T$ is a positive definite matrix and it is thus invertible. Hence, the optimal solution of $\mathbf{w}$ is

$$\mathbf{w} = \left(\sum_{i\in\mathcal{I}}\mathbf{x_i}\mathbf{x_i}^T\right)^{-1}\left(\sum_{i\in\mathcal{I}} y_i\mathbf{x_i} - \frac{1}{2}\mathbf{C}^T\mathbf{A}\right). \tag{14}$$

3.3.4 Proofs of Convergence

Each of the alternating optimization steps decreases the objective function Ψ, and the objective function has a lower bound 0. We thus can ensure that the convergence of the alternating optimization.

4 Online Search

For a newly posted question, we can search the repositories to find its similar questions. There exist many proven techniques in finding the top k similar questions, such as Cosine similarity, syntactic tree matching approach [10], and other representation learning based methods. In this work, we employed the Apache Lucene[9]-based k-NN strategy to find the top k similar questions. Herewith the returned k similar

8. http://tinyurl.com/zvd8zte

9. https://lucene.apache.org/

TABLE 2
Statistics of Preference Pairs and Selected Testing Samples

cQA Site	Preference Pairs					Selected Testing Samples ($P@K$, 10 times)
	Question #	Answer #	Pos. Pref. Pair #	Neg. Pref. Pair #	Neu. Pref. Pair #	Total Question #
HealthTap	1,959	6,903	27,426	27,426	4,785	1,000
Zhihu.com	8,241	30,619	123,615	123,615	32,964	1,000

'Pos.', 'Pref.', 'Neg.', and 'Neu.' are short for 'positive', 'preference', 'negative', and 'neutral', respectively.

questions, we can easily construct an answer candidate pool by gathering all the answers associated to the k returned questions. We then pair the given question with each of the answers in the pool. Following that, we utilize our model to generate an answer ranking list by pairwise comparison.

The number of the paired QAs is very small, since k is very small (usually less than 10). Therefore, we can efficiently extract their features and judge their preference relationships on line.

5 Experiments

5.1 Experimental Settings

To train our proposed PLANE model, we have to first construct the preference pairs. Towards this end, we randomly selected approximately two and eight thousand questions and their associated answers from our collected HealthTap and Zhihu.com datasets, respectively. Each of these selected questions were required to receive at least three answers. We then built the training set $\mathcal{X}$ with balanced positive and negative preference pairs. Meanwhile, we built the training set $\mathcal{U}$ with neutral preference pairs. The statistics are summarized in Table 2. The remaining questions and their answers in each dataset serve as the whole testing sets.

To thoroughly justify our model, we randomly selected 100 questions from each of the testing set (excluding those for training) and repeated 10 times. For each of the selected testing question q, we first performed k-NN ($k = 5$) to search its similar questions over the whole testing set. Undoubtedly, the question itself can be found. We then constructed an answer candidate pool by gathering all the answers associated to the selected questions. In the light of this, we can ensure that the real answer/answers of the given testing question is/are in the answer candidate pool. Following that, we paired the given question with each of the answers in the pool. Hereafter, we utilized our model to generate an answer ranking list by pairwise comparison.

Regarding answer selection, precision is more important than recall. We thus measured our model via the widely-accepted metric, average $P@K$ [43]. A ranking list was generated for each given question. We adopted $P@K$ to measure the ranking performance of each list. We define $P@K = \frac{|\mathcal{C} \cap \mathcal{T}|}{|\mathcal{C}|}$, where $\mathcal{C}$ is a set of the top K answers in the ranking list, and $\mathcal{T}$ is the set of the true ones in $\mathcal{C}$. The true ones refer to the answers that are the real answers to the given question. $P@K$ stands for the proportion of the selected answers in the top K that are true.

We, in fact, indolently established the ground truth without any labeling efforts. Nonetheless, it is reasonable due to the following reasons. First, for some professional questions, such as the questions from HealthTap, only the experts with specific knowledge are qualified to judge the matching degree between a question and an answer. It is thus very hard, if not impossible, to find such annotators who are professional across all the fields. Second, the strategy of our current ground truth construction ensures that there exist at least one relevant answer in the answer candidate pool for the given question. In addition, we would like to clarify that we did not perform the 10-fold cross-validation, because the validation is on the ranking results instead of the preference pair classification.

All the experiments were conducted over a server equipped with Intel(R) Core(R) CPU i7-4790 at 3.60 GHZ on 32 GB RAM, four cores, and 64-bit Windows 10 operating system.

5.2 Feature Extraction

To comprehensively represent each QA pair, we extracted several feature types:

Deep Features. We adopted a Doc2Vec method, which is also known as para2vec or sentence embedding [44]. This method modifies the word2vec algorithm to unsupervised learning of fixed-length and continuous representations for larger blocks of texts. The texts can be of variable-length, ranging from sentences, paragraphs, to entire documents. With the help of this publicly accessible tool,[10] we set the sliding window size as 5 and the dimension of the representation as 50. We leveraged our collected QA pairs from HealthTap and Zhihu.com to train the Doc2Vec model, respectively. Compared to the traditional unigram features, the extracted deep features can capture the semantics and orderings of words. It hence well characterizes the contexts of the QA pairs.

Topic-Level Features. We also employed the Latent dirichlet allocation (LDA) based topic-level features for QA representation [45]. In particular, each latent topic was deemed as one feature. The number of topics was tuned according to the widely-adopted perplexity metric [45]. Regarding perplexity, a lower value usually indicates a better LDA model. We divided the QA pairs into two subsets: 80 percent was used to train the LDA models with various numbers of latent topics; and 20 percent was used for evaluation in terms of perplexity. The LDA model and the perplexity metric were implemented with the help of the Stanford topic modeling toolbox.[11] In our work, when the number of latent topics arrives at 50, the perplexity curve reaches the trough. Each QA pair was hence represented as a 50 dimensional semantic feature vector. The topic-level features are capable of capturing the high-level semantics of QA pairs with lower dimensionality.

10. https://github.com/klb3713/sentence2vec
11. http://nlp.stanford.edu/software/tmt/tmt-0.4/

Statistical Features. This feature set consists of the statistics of the surface literature. It captures the number of prominent countable terms and surface commonality of a question and its answer. We argue that a good answer has a good structure and contains a reasonable number of syntactic features. In particular, we first independently extracted features from the question and answer pair, including the number of terms, verbs, nouns, tags, and stop words of the questions and answers, respectively. Also, we extracted the number of overlapped terms between questions and answers.

User-Centric Features. To justify the responsibility, activity, willingness, and reputation of users, we extracted a set of user-centric features. In particular, we considered the users' biography length, working years, the number of followers, followees, received thanks, received agree, asked questions, answered questions, log edit, and followed questions. It is worthy that the roles of askers and answerers in the general cQA services, such as Zhihu.com, are exchangeable and publicly accessible; while they are not in the vertical cQA, such as HealthTap. Meanwhile, in vertical cQA, the askers' profiles are inaccessible due to the privacy issues. Hence, some user-centric features are not applicable in HealthTap.

5.3 Performance Comparison with Baselines

To demonstrate the effectiveness of our proposed method, i.e., PLANE, we compared it with the following state-of-the-art answer selection baselines:

- RF: To rank answers in QA forums, a *pointwise* learning to rank method with Random Forests (RF) was proposed by Dalip et al. [24]. In particular, given a set of training samples $\{(\mathbf{x_1}, r_1), \ldots, (\mathbf{x_n}, r_n)\}$, where $\mathbf{x_i}$ denotes the feature vector of the ith QA, and r_i refers to their relevance, i.e., number of votes in our work. The RF method works by training a predictor $T(\cdot)$ such that $T(\mathbf{x_i}) \sim r_i$.
- LR: Logistic regression, LR for short, is a binary classification method. It outputs a score referring to a probability of relevance, and it is hence a natural as well as effective choice for question-answering tasks [25]. In our experiments, we treated the pair of (question, best answer) as a positive sample and the pair of (question, non-best answer) pair as a negative one. This method is a *pointwise* approach.
- GBRank: This *pairwise* model sorts answer candidates by learning a ranking function h over a given set of preferences, such that $h(\mathbf{x_i}) \geq h(\mathbf{x_j})$, if $\mathbf{x_i} \succ \mathbf{x_j}$. A squared hinge loss function was used to measure the risk of h formulated as $\frac{1}{2}\sum_i \max(0, h(\mathbf{x_i}) - h(\mathbf{x_j}) + \tau)$. Bian et al. [26] solved h using gradient boosting (GBRank). In the implementation, only answer pairs under the same question were utilized for training.
- RankSVM: The Ranking SVM (RankSVM) algorithm is a learning retrieval function that employs *pairwise* ranking methods to adaptively sort results based on how 'relevant' they are to a specific query [28]. The original purpose of the algorithm was to improve the performance of an internet search engine. Hieber et al. [27] used RankSVM to improve the performance of answer ranking in social QA portals and achieved promising performance. Similar as GBRank, the model was trained with the answer pairs under the same question.
- AdaRank: Unlike the pointwise or pairwise methods where the loss functions over the individual answer candidate or a pair are minimized loosely related to the performance metrics, AdaRank [30] is a *listwise* approach. It is capable of minimizing a loss function directly on the performance metrics, ($P@10$ in this work).
- Three-classes: We treated the answer selection task as a three-category classification problem (positive, neutral, and negative preferences). We trained a SVM classifier with the radial basis function kernel. This one is close to ours.

For each method mentioned above, the involved parameters were carefully tuned, and the parameters with the best performance were used to report the final comparison results. We will detail the parameter tuning and sensitivity analysis in the next section. In addition, we have to clarify that the relative preference pairs for all the baselines (except the last one) only include the answers to the current questions (Observation 1 only). This is reasonable in our experimental settings, since the observation-guided training samples building is part of our whole model and the last two observations are never used before.

The experimental results of our proposed PLANE model and five baselines are summarized in Table 3. From this table, we can observe the following points: 1) The pointwise approaches, i.e., RF and LR, achieved the worst performance regarding $P@K$ with different depth. Pointwise approaches transform ranking into classification or regression on single QA pairs. They are thus unable to consider the relative orders (preference) between two answers. Nevertheless, ranking is more about predicting relative orders of answers rather than precise relevance scores. That is why their performance is comparatively suboptimal. 2) It is obvious that the pairwise baselines, i.e., GBRank, RankSVM, and our proposed PLANE, outperform pointwise approaches. This is because the pairwise approaches minimize the number of pairs which are ranked out of order, and they thus model the preference relationship between any two QA pairs rather than the absolute value of the relevance degree of a QA pair. 3) The listwise approach, namely AdaRank, has its advantages as compared to the traditional pairwise approaches excluding our proposed PLANE. This is because pairwise ranking algorithms often do not consider the position of answers in the final ranking results, but instead define their loss functions directly on the basis of individual preference pairs. By contrast, the listwise approach takes the entire ranking list and minimizes measure-specific loss functions. 4) Our proposed model is stably and substantially better than the traditional pointwise, pairwise, and listwise baselines. This is caused by several reasons. First, it considers the pairwise preference that is why it is superior to the pointwise baselines. Second, in addition to the positive and negative preference pairs, it also incorporates the neutral preference pairs. This results in its superior to the traditional pairwise and listwise approaches. It also signals that the neutral preference pairs convey much valuable information. And 5) compared to the three-classes, its ℓ_1 norm constraint enables the feature selection. This reflects that not all the extracted

TABLE 3
Performance Comparison between Our PLANE Model and Several State-of-the-Art Baselines over Two Data Sets

Data Sets	Methods	P@1	P@2	P@3	P@4	P@5	P-value
HealthTap Data Set	RF (*pointwise*)	$13.3 \pm 2.45\%$	28.4 ± 4.45 %	$42.6 \pm 4.25\%$	$61.1 \pm 4.21\%$	$75.7 \pm 3.52\%$	$8.15e$-06
	LR (*pointwise*)	$15.7 \pm 3.72\%$	$30.3 \pm 4.44\%$	$45.9 \pm 6.08\%$	$61.2 \pm 5.66\%$	$77.4 \pm 4.34\%$	$6.52e$-04
	GBRank (*pairwise*)	$16.8 \pm 4.09\%$	$33.0 \pm 4.58\%$	$46.9 \pm 3.54\%$	$63.3 \pm 5.75\%$	$77.9 \pm 3.68\%$	$2.80e$-04
	RankSVM (*pairwise*)	$17.3 \pm 2.79\%$	$31.2 \pm 3.92\%$	$46.3 \pm 3.72\%$	$58.7 \pm 5.16\%$	$70.9 \pm 4.47\%$	$3.39e$-04
	AdaRank (*listwise*)	$23.9 \pm 2.69\%$	$41.0 \pm 4.15\%$	$55.9 \pm 5.41\%$	$70.2 \pm 4.75\%$	$83.6 \pm 3.13\%$	$4.18e$-02
	3-classes	$32.2 \pm 2.93\%$	$50.1 \pm 3.67\%$	$63.6 \pm 3.95\%$	$75.4 \pm 3.83\%$	$85.6 \pm 3.72\%$	$2.22e$-02
	PLANE (*our model*)	$33.2 \pm 3.52\%$	$52.6 \pm 3.67\%$	$64.6 \pm 4.13\%$	$76.2 \pm 4.14\%$	$86.3 \pm 3.80\%$	- -
Zhihu.com Data Set	RF (*pointwise*)	$33.6 \pm 3.23\%$	$51.7 \pm 4.96\%$	$63.0 \pm 4.49\%$	$72.8 \pm 4.12\%$	$83.1 \pm 3.18\%$	$1.18e$-04
	LR (*pointwise*)	$32.7 \pm 4.94\%$	$44.7 \pm 4.41\%$	$57.3 \pm 4.43\%$	$68.1 \pm 3.72\%$	$81.8 \pm 2.72\%$	$1.23e$-07
	GBRank (*pairwise*)	$35.5 \pm 3.07\%$	$46.1 \pm 6.50\%$	$61.2 \pm 6.29\%$	$74.0 \pm 5.18\%$	$86.2 \pm 3.97\%$	$2.97e$-04
	RankSVM (*pairwise*)	$35.2 \pm 2.93\%$	$55.3 \pm 2.41\%$	$70.4 \pm 4.13\%$	$82.5 \pm 2.54\%$	$90.1 \pm 2.21\%$	$1.28e$-04
	AdaRank (*listwise*)	$35.7 \pm 5.12\%$	$56.7 \pm 3.77\%$	$71.0 \pm 3.69\%$	$83.1 \pm 3.36\%$	$91.7 \pm 3.03\%$	$1.79e$-02
	3-classes	$46.4 \pm 4.08\%$	$64.6 \pm 3.76\%$	$76.9 \pm 4.01\%$	$85.3 \pm 4.38\%$	$91.4 \pm 1.85\%$	$1.21e$-02
	PLANE (our model)	$47.6 \pm 3.53\%$	$68.3 \pm 3.10\%$	$77.7 \pm 4.82\%$	$86.3 \pm 4.41\%$	$92.5 \pm 2.38\%$	- -

It is measured by $P@K$ with different depth. We also provide the variance. Significance test is based on P@5.

features are discriminative and feature selection benefits the classification performance. It is worth emphasizing that other baselines do not have the capability of feature selection.

We also conducted the analysis of variance (known as ANOVA) based on $P@5$. In particular, we performed paired t-test between our model and each of the baselines over the 10-round results. The results are displayed in the last column of Table 3. We found that all the p-values are substantially smaller than 0.05, which shows that the improvements of our proposed model are statistically significant.

5.4 Component-Wise Evaluation

There are three key observations guiding us to build three kinds of preference pairs. We conducted experiments to verify their effectiveness one by one:

1) Drop 1: For a given question, we did not consider the preference pairs between its best and non-best answers.
2) Drop 2: For a given question, we did not consider the neutral preference pairs between its non-best answers.
3) Drop 3: For a given question, we did not consider the preference pairs between its own and the answers of other questions.
4) PLANE: It is our current work which considers all the preference pairs.

The results are displayed in Table 4. It can be seen that: 1) No matter what type of preference pairs we dropped does hurt the performance of our model. This verifies the importance and necessity of these three kinds of preference pairs. 2) "Drop 1" achieves the worst performance. This signals that the preference pairs between best versus non-best answers have a major influence on the overall performance. And 3) it is clear that the neutral preference pairs do have contributions to the overall performance. That is why our model outperforms other pairwise learning to rank models. It is notable that the performance of "Drop 2" in Table 4 is much better than GBRank and RankSVM in Table 3, even all these three are pairswise methods. This is because that traditional pairwise ranking methods for answer ranking only considers the answer pairs under the same question. Within our third observation, more reasonable answer pairs are fed into the model for training.

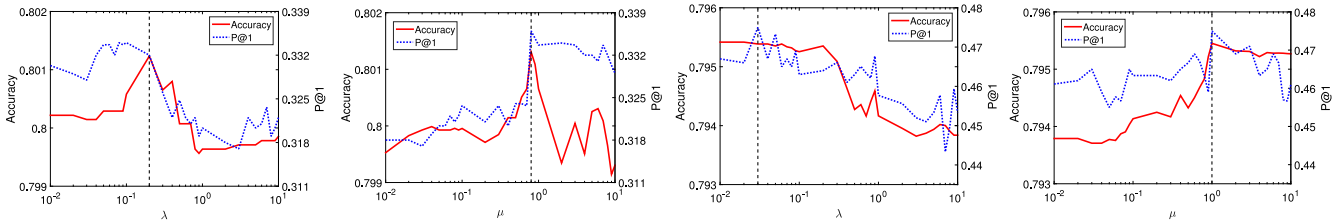
5.5 Parameter Tuning and Sensitivity Analysis

We have two key parameters λ and μ as shown in Eqn. (4). The optimal settings of these parameters were carefully tuned on the HealthTap and Zhihu.com datasets, separately. In particular, we leveraged all the constructed preference pairs as summarized in Table 2 to train our model and the 10 (times) $\times$ 100 (testing questions) to validate our model. Grid search was employed to select the optimal parameters between 10^{-2} and 10^{2} with small but adaptive step sizes. The step sizes were 0.01, 0.1, and 1 for the range of [0.01, 0.1], [0.1, 1], and [1, 10], respectively. The parameters corresponding to the best average $P@1$ were used to report the final results. For other baselines, the procedures to tune the parameters are analogous to ensure a fair comparison.

Beside the $P@K$ metric for answer selection performance, we studied the parameter tuning regarding the preference pair classification, which is the middle result of the

TABLE 4
Component-Wise Evaluation by Removing One of the Three Kinds of Preference Pairs Each Time

Measure	HealthTap Data Set				Zhihu.com Data Set			
	Drop 1	Drop 2	Drop 3	PLANE	Drop 1	Drop 2	Drop 3	PLANE
P@1	$30.3 \pm 3.42\%$	$32.7 \pm 3.29\%$	$32.8 \pm 3.52\%$	$33.2 \pm 3.52\%$	$44.7 \pm 3.21\%$	$46.3 \pm 3.22\%$	$46.2 \pm 3.59\%$	$47.6 \pm 3.52\%$
P@2	$47.0 \pm 3.82\%$	$51.9 \pm 3.46\%$	$50.9 \pm 3.62\%$	$52.6 \pm 3.67\%$	$65.3 \pm 3.52\%$	$67.5 \pm 3.42\%$	$66.4 \pm 3.13\%$	$68.3 \pm 3.10\%$
P@3	$60.3 \pm 3.91\%$	$63.7 \pm 3.92\%$	$63.1 \pm 4.41\%$	$64.6 \pm 4.13\%$	$77.0 \pm 4.72\%$	$77.2 \pm 3.62\%$	$69.8 \pm 3.60\%$	$77.7 \pm 4.81\%$
P@4	$74.3 \pm 4.17\%$	$75.2 \pm 3.54\%$	$74.6 \pm 4.73\%$	$76.2 \pm 4.14\%$	$83.7 \pm 4.28\%$	$85.1 \pm 3.53\%$	$84.3 \pm 3.25\%$	$86.3 \pm 4.41\%$
P@5	$85.2 \pm 4.62\%$	$86.1 \pm 3.98\%$	$85.3 \pm 3.15\%$	$86.3 \pm 3.80\%$	$90.4 \pm 2.11\%$	$91.8 \pm 2.89\%$	$91.5 \pm 2.96\%$	$92.5 \pm 2.38\%$



(a) Tuning λ on HealthTap (b) Tuning μ on HealthTap (c) Tuning λ on Zhihu.com (d) Tuning μ on Zhihu.com

Fig. 3. Parameter tuning on HealthTap and Zhihu.com with two different metrics. Grid search strategy with adaptive step sizes was employed to find the optimal parameters.

answer selection target. The performance of preference pair classification is measured by accuracy. To be more specific, we split the constructed preference pairs into three chunks: 80 percent of the preference pairs were used for training, 10 percent were used for validation, and the rest were held-out for testing. The training set was used to adjust the parameters, while the validation set was used to minimize overfitting, i.e., verifying that any performance increase over the training dataset actually yields an accuracy increase over the dataset that has not been shown to the model before. The testing set was used only for testing the final solution to confirm the actual predictive power of our model with optimal parameters. We also employed the same grid search strategy to tune the two parameters.

Figs. 3a and 3b illustrate the performance of our model with respect to parameters λ and μ on HealthTap. Figs. 3c and 3d illustrate that on Zhihu.com. The figures were drawn by fixing one parameter and varying the other. From these four figures, we have the following observations: 1) The curves of $P@1$ and accuracy have the similar trends. That further indicates that the performance of our proposed PLANE model has an immediate impact on the answer selection performance. And 2) the performance of our proposed PLANE model changes within small ranges nearby the optimal settings, even the two parameters vary in a relatively wide range. This justifies that our model is non-sensitive to the parameters around their optimal settings.

5.6 Robustness Validation

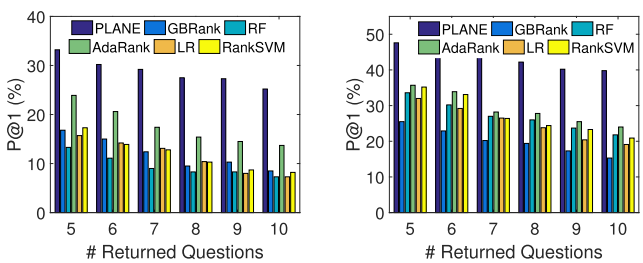
One key stage in our work is to use the k-NN method to find the similar questions for the given one. In our current experimental settings, even k is set as 1, we can still ensure that the "matched" answers fall into the answer candidate pool, because the given question itself can be positioned at the first place based on the k-NN algorithm. However, in the real-world settings, for a given question of interest, there is no earthly chance to find the exactly matched questions from the archive. We thus have to enlarge k to improve the recall of similar questions and hence the "matched answers". But say, a larger k may introduce more noises into the answer candidate pool in the form of irrelevant answers, and it thus increases the difficulty of the answer selection task.

To validate the robustness of our proposed PLANE model and the baselines, we varied the number of the returned similar questions, i.e., k in k-NN, on two datasets, respectively. Specifically, we varied k from 5 to 10 and measured the performance via average $P@1$ of each approach over each dataset. The results are shown in Figs. 4a and 4b. Jointly analyzing these two figures, we can make the following observations: 1) The overall performance trends of all the models decrease as k increases. This confirms our concern that larger k settings bring in more noises and hence incur bigger challenges. 2) The performance of all the models on HealthTap decreases faster than that on Zhihu.com. The reason for this phenomenon may be that questions from vertical cQA sites are, more often than not, very specific, complex, and sophisticated, which orient personalized and professional answers. Therefore, there exist only a few questions that are very similar to the given one. As compared to the general cQA sites, the vertical ones have more noises even under the same k settings. 3) The advantage of our proposed model is much more obvious on HealthTap as compared to that on Zhihu.com. This reveals that our model performs better under the nosy contexts, and it is hence much more robust. And 4) the performance of our model is consistently better than that of the baselines. Meanwhile, the performance drop with increasing k is not very large. This further justifies the robustness of our model.

5.7 Alternatives of Preference Pair Construction

As introduced in Section 3.1, in our current preference pair construction process, guided by the second observation, we assign equal quality to all non-best answers under the same question. Apart from that, we also explored two alternatives and verified their effectiveness:

- Original: This is the original method introduced in Section 3.1.
- Absolute: This alternative is to build preference pairs based on their votes. In particular, for a given question q_i, we have $(q_i, a_i^j) \succ (q_i, a_i^k)$ if the jth answer obtains more votes than the kth answer and vice versa. This approach outputted the same amount of training examples. But the number of neutral preference pairs is significantly reduced.
- Threshold: Another strategy is the threshold-based construction. Specifically, we set a pre-defined threshold T. We have $(q_i, a_i^j) \cong (q_i, a_i^k)$ if the vote



(a) Performance on HealthTap (b) Performance on Zhihu.com

Fig. 4. Performance comparison among different models w.r.t. varying the number of returned similar questions, i.e., varying the k value in the k-NN model.

TABLE 5
Comparison between the "Original" and "Absolute" Preference Pair Construction Methods on Two Datasets

Measure	HealthTap Data Set			Zhihu.com Data Set		
	Absolute	Original	P-value	Absolute	Original	P-value
P@1	$32.3 \pm 3.42\%$	$33.2 \pm 3.52\%$	$5.25e$-05	$43.0 \pm 3.21\%$	$47.6 \pm 3.53\%$	$6.42e$-04
P@2	$50.6 \pm 3.54\%$	$52.6 \pm 3.67\%$	$8.32e$-04	$59.3 \pm 3.15\%$	$68.3 \pm 3.10\%$	$5.26e$-05
P@3	$62.8 \pm 4.58\%$	$64.6 \pm 4.13\%$	$2.68e$-05	$67.5 \pm 4.21\%$	$77.7 \pm 4.81\%$	$4.73e$-05
P@4	$74.8 \pm 3.76\%$	$76.2 \pm 4.14\%$	$5.27e$-05	$74.1 \pm 4.33\%$	$86.3 \pm 4.41\%$	$7.24e$-05
P@5	$84.3 \pm 4.21\%$	$86.3 \pm 3.80\%$	$2.94e$-05	$78.8 \pm 2.53\%$	$92.5 \pm 2.38\%$	$3.64e$-06

difference between the jth and kth answers of the ith question is not greater than T; otherwise, we give up the pairs. We increase T to generate more neutral preference pairs. In fact, the "Original" method are the special cases of the "Threshold" one. When T tends to be infinite, "Threshold" becomes the "Original" method.

The comparison results between "Original" and "Absolute" methods are summarized in Table 5. We observed the following points: 1) The performance based on "Original" method is consistently better than that of the "Absolute" one, on both data sets. The possible reason may be that the "Absolute" method brings in a higher level of noise, since there are large chances that a worse answer receives a few more votes. As discussed in Section 5.8, the untruthful votes of an answer would incurs a series of incorrect preference pairs, especially in the "Absolute" method. And 2) the advantage of "Original" method over the "Absolute" one is much more remarkable over the Zhihu.com data set as compared to that over the HealthTap data set. This reflects that the non-best answers in the general QA sites are harder to differ than those in the domain-specific QA sites.

The validating results of the "Threshold" approach for preference pair construction is illustrated in Fig. 5. It is clear that the performance in terms of P@K with various depth goes up first and then tends to be stable as T increases. As discussed before, the "Original" method equals to the "Threshold" one at $T = \infty$. This reveals that the "Original" method achieves the optimal performance. Also, we can see that when T equals to zero, the results are better than that of the "Absolute" method, especially on the Zhihu.com data set. This again tells us that the number of votes on the non-best answers are not very reliable.

5.8 Outlier Case Study

As detailed in Section 3.1, the strategies of building our training samples (i.e., preference pairs) were guided by our observations from a statistical view. We thus cannot ensure that there are no outlier cases distant from our overall observations. Since the number of preference pairs can be up to the quadratic order of the number of QA pairs, the pairwise approach is more sensitive to error labels than the pointwise approach. For example, suppose that a question q has four answers, and one of its non-best answers happen to be better than its best answer. In such context, we will produce a series of wrong preference pairs and it will greatly hinder the learning performance. Consequently, the outlier cases deserve our special attention.

To vividly demonstrate the outlier cases, we selected three examples from HealthTap and Zhihu.com, respectively. They are displayed in Figs. 7a and 7b. These three examples cover all the cases against our observations introduced in Section 3.1:

1) From the left column, it can be seen that, for a given question, the answers of its similar questions receive higher votes and are much more descriptive than its own. This demonstrates the special cases that the answers of other questions may be better than its own.
2) From the middle column, it can be seen that, for a given question, the quality of its some non-best answers in terms of informativeness and relevance are substantially better than the other non-best ones. These cases also break our observation conclusion.
3) From the right column, we can notice that the non-best answers of a given question may provide more informative cues than its best one.

Detecting and removing the outlier cases before building the preference pairs will remarkably boost the learning performance of our proposed PLANE model and other pairwise learning to rank models. We will focus on this research direction in the future.

5.9 Complexity Analysis

To analyze the complexity of our proposed PLANE model, we need to solve the time complexity in the computation

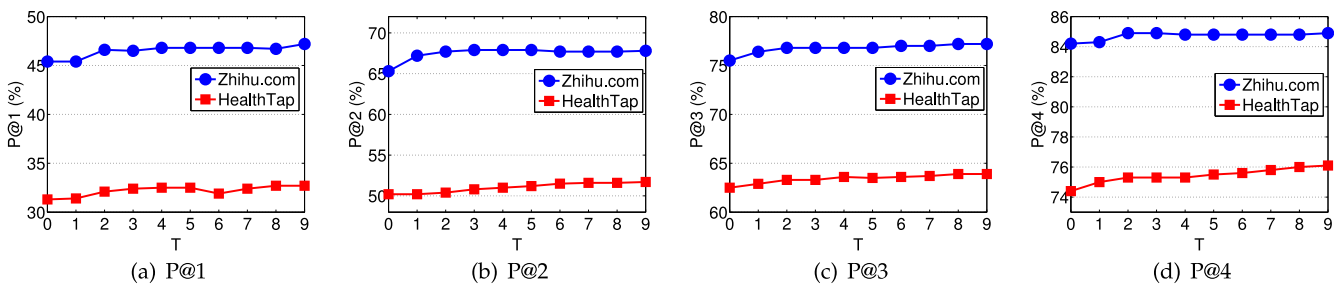
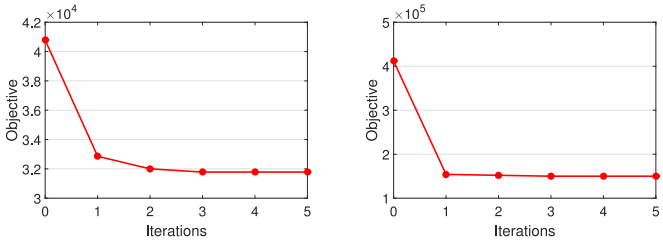


Fig. 5. Validation of the "Threshold" approach for preference pair construction. It is measured by P@k w.r.t the threshold T variant. We can see that the performance tends to be stable as T increases.



(a) PLANE on HealthTap (b) PLANE on Zhihu.com

Fig. 6. Convergence process illustration of PLANE model on HealthTap and Zhihu.com, respectively. We can see that it converges very fast.

of $\mathbf{A}$ and $\mathbf{w}$ as defined in Eqns. (10) and (14), respectively. The computation of Matrix $\mathbf{A}$ has a time complexity of $O_A = O(D(D+M))$. The number of elements in $\mathcal{I}$ is less than N, and we assume $N \gg D$, so the computation cost of $\mathbf{w}$ is $O_w = O(D^2N)$. Therefore, the complexity of PLANE model is $O(T(O_A + O_w)) = O(TD^2N)$, where T indicate the number of iterations in the training process. In practice, the PLANE model converges very fast. As shown in Figs. 6a and 6b, the optimal solution of the PLANE model can be reached within less than five iterations.

6 Conclusion and Future Work

In this work, we present a novel scheme for answer selection in cQA settings. It comprises of an offline learning and an online search component. In the offline learning component, instead of time-consuming and labor-intensive annotation, we automatically construct the positive, neutral, and negative training samples in the forms of preference pairs guided by our data-driven observations. We then propose a robust pairwise learning to rank model to incorporate these three types of training samples. In the online search component, for a given question, we first collect a pool of answer candidates via finding its similar questions. We then employ the offline learned model to rank the answer candidates via pairwise comparison. We have conducted extensive experiments to justify the effectiveness of our model on one general cQA dataset and one vertical cQA dataset. We can conclude the following points: 1) our model can achieve better performance than several state-of-the-art answer selection baselines; 2) our model is non-sensitive to its parameters; 3) our model is robust to the noises caused by enlarging the number of returned similar questions; 4) the pairwise learning to rank models including our proposed PLANE are very sensitive to the error training samples.

Beyond the traditional pairwise learning to rank models, our model is able to incorporate the neutral training samples and select the discriminative features. It, however, also has the inherent disadvantages of the pairwise learning to rank family, such as noise-sensitive, large-scale preference pairs, and loss of information about the finer granularity in the relevance judgment. In the future, we plan to address such disadvantages in the field of cQA.

Question: Hi... Is there any way to get rid of snoring.. My mom has a serious problem of snoring.	**Question**: Do lots of people get asthma, or is it unusual?	**Question**: Abdominal discomfort above navel, feels like pressure, could it be a hernia? I have had a CT scan and blood work, both normal.
Ans.: Suggest she see a pulmonologist. Snoring can be a sign of sleep apnea. There are effective treatments for this issue. (0 vote)	**Non-Best Ans.1**: Asthma is one of the most common chronic conditions in childhood, affecting 9% of the pediatric population. As many as 30% of children will have wheezing, not all of which is due to asthma. Asthma is often missed in children. (1 vote)	**Best Ans.**: Yes It could be a hernia. An epigrams trick type hernia. Do you have a palpable bump or knit in that area? (6 votes)
Similar Question: How to stop my snoring?	**Non-Best Ans.2**: Common... Asthma is actually quite common. (0 vote)	**Non-Best Ans.**: It can be anything. Naproxen can cause gastritis. Also, heartburn, pregnancy, spasms from period. The list is extensive because there's many organs in the abdominal area. So the best approach would be to see your doctor, and have this examined closely. Once it's diagnosed, it will be easy to treat. (3 votes)
Ans.: Sleep on your side If you are a mild snorer, sleeping on your side may be all you need to do. A bumper belt can help. If your snoring is more vigorous, you may have sleep apnea. You need to see your doctor who can order a sleep study. If you have sleep apnea options ... (5 votes)	**Non-Best Ans.3**: Difficult to answer. Asthma is certainly increasing in prevalence, as are allergic diseases in general. I would not classify asthma as being "unusual". Risk factors for developing asthma include ... (0 vote)	

(a) Illustrative examples selected from HealthTap

Question: Is there any means of keeping fit, if I prefer to stay at home?	**Question**: Did photographers care the theoretical knowledge in shootings?	**Question**: "HTC One S" and "Sansung Galaxy Nexus", which is better?
Ans.: Dumbbells are good for you. Take exercises according to videos. (2 votes)	**Non-Best Ans.1**: Poor light, give up; poor composition, give up. (0 vote)	**Best Ans.**: I recommend Sony lt26i and Meizu M. If I have to choose one in from "HTC One S" and "Samsung GN", I prefer the latter one. (1 votes)
Similar Question: Is there any simple method to make exercises indoors?	**Non-Best Ans.2**: It is a great shame to check books. (0 vote)	
Ans.: Dumbbells and Pilates mats are recommended. Do 6-15 push-ups, 8-10 dumbbell press ... But you should be careful at the beginning, and it is important to warm up. (31 votes)	**Non-Best Ans.3**: If sufficient practices are taken, the composition will be not bad without considering too much. As to the light, they usually take a test and modify the settings. (0 vote)	**Non-Best Ans.**: "One S". In my opinion, most Android phones look similar in appearance, so I prefer the one with unique look. Moreover, the GPU and CPU in "One S" are better. (0 votes)

(b) Illustrative examples selected from Zhihu.com

Fig. 7. Outlier case study. In the left column, we intend to show the special cases that for a given question, the answers of other questions may be better than its own. In the middle column, we demonstrate the exceptional cases that for a given question, some of its non-best answers may be significantly better than the other non-best answers. In the right one, we aim to explain the special scenarios, whereby the non-best answers of a given question may provide more informative cues than its best one.

REFERENCES

[1] M. Ali, M. Li, W. Ding, and H. Jiang, *Modern Advances in Intelligent Systems and Tools*, vol. 431. Berlin, Germany: Springer, 2012.

[2] L. Nie, M. Wang, L. Zhang, S. Yan, B. Zhang, and T. S. Chua, "Disease inference from health-related questions via sparse deep learning," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 8, pp. 2107–2119, Aug. 2015.

[3] A. Shtok, G. Dror, Y. Maarek, and I. Szpektor, "Learning from the past: Answering new questions with past answers," in *Proc. 21st Int. Conf. World Wide Web*, 2012, pp. 759–768.

[4] E. Agichtein, C. Castillo, D. Donato, A. Gionis, and G. Mishne, "Finding high-quality content in social media," in *Proc. Int. Conf. Web Search Data Mining*, 2008, pp. 183–194.

[5] J. Jeon, W. B. Croft, J. H. Lee, and S. Park, "A framework to predict the quality of answers with non-textual features," in *Proc. 29th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2006, pp. 228–235.

[6] Z. Ji and B. Wang, "Learning to rank for question routing in community question answering," in *Proc. 22nd ACM Int. Conf. Inf. Knowl. Manage.*, 2013, pp. 2363–2368.

[7] T. C. Zhou, M. R. Lyu, and I. King, "A classification-based approach to question routing in community question answering," in *Proc. 21st Int. Conf. World Wide Web*, 2012, pp. 783–790.

[8] L. Yang, et al., "CQArank: Jointly model topics and expertise in community question answering," in *Proc. 22nd ACM Int. Conf. Inf. Knowl. Manage.*, 2013, pp. 99–108.

[9] B. Li and I. King, "Routing questions to appropriate answerers in community question answering services," in *Proc. 19th ACM Int. Conf. Inf. Knowl. Manage.*, 2010, pp. 1585–1588.

[10] K. Wang, Z. Ming, and T.-S. Chua, "A syntactic tree matching approach to finding similar questions in community-based QA services," in *Proc. 32nd Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2009, pp. 187–194.

[11] Y. Liu, J. Bian, and E. Agichtein, "Predicting information seeker satisfaction in community question answering," in *Proc. 31st Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2008, pp. 483–490.

[12] M. J. Blooma, A. Y. K. Chua, and D. H.-L. Goh, "A predictive framework for retrieving the best answer," in *Proc. ACM Symp. Appl. Comput.*, 2008, pp. 1107–1111.

[13] L. Nie, M. Wang, Y. Gao, Z. Zha, and T. Chua, "Beyond text QA: Multimedia answer generation by harvesting Web information," *IEEE Trans. Multimedia*, vol. 15, no. 2, pp. 426–441, Feb. 2013.

[14] Q. H. Tran, V. D. Tran, T. T. Vu, M. L. Nguyen, and S. B. Pham, "JAIST: Combining multiple features for answer selection in community question answering," in *Proc. 9th Int. Workshop Semantic Eval.*, 2015, pp. 215–219.

[15] W. Wei, et al., "Exploring heterogeneous features for query-focused summarization of categorized community answers," *Inf. Sci.*, vol. 330, pp. 403–423, 2016.

[16] S. Tellex, B. Katz, J. Lin, A. Fernandes, and G. Marton, "Quantitative evaluation of passage retrieval algorithms for question answering," in *Proc. 26th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2003, pp. 41–47.

[17] H. Cui, R. Sun, K. Li, M.-Y. Kan, and T.-S. Chua, "Question answering passage retrieval using dependency relations," in *Proc. 28th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2005, pp. 400–407.

[18] R. Sun, H. Cui, K. Li, M.-Y. Kan, and T.-S. Chua, "Dependency relation matching for answer selection," in *Proc. 28th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2005, pp. 651–652.

[19] M. Surdeanu, M. Ciaramita, and H. Zaragoza, "Learning to rank answers on large online QA collections," in *Proc. 46th Annu. Meeting Assoc. Comput. Linguistics: Human Language Technol.*, 2008, pp. 719–727.

[20] A. Agarwal, et al., "Learning to rank for robust question answering," in *Proc. 21st ACM Int. Conf. Inf. Knowl. Manage.*, 2012, pp. 833–842.

[21] D. Savenkov, "Ranking answers and Web passages for non-factoid question answering: Emory university at TREC LiveQA," in *Proc. 24th Text REtrieval Conf.*, 2015.

[22] X. Li, Y. Ye, and M. K. Ng, "MultiVCRank with applications to image retrieval," *IEEE Trans. Image Process.*, vol. 25, no. 3, pp. 1396–1409, Mar. 2016.

[23] M. K. Ng, X. Li, and Y. Ye, "MultiRank: Co-ranking for objects and relations in multi-relational data," in *Proc. 17th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2011, pp. 1217–1225.

[24] D. H. Dalip, M. A. Gonçalves, M. Cristo, and P. Calado, "Exploiting user feedback to learn to rank answers in QA forums: A case study with stack overflow," in *Proc. 36th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2013, pp. 543–552.

[25] C. Shah and J. Pomerantz, "Evaluating and predicting answer quality in community QA," in *Proc. 33rd Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2010, pp. 411–418.

[26] J. Bian, Y. Liu, E. Agichtein, and H. Zha, "Finding the right facts in the crowd: Factoid question answering over social media," in *Proc. 17th Int. Conf. World Wide Web*, 2008, pp. 467–476.

[27] F. Hieber and S. Riezler, "Improved answer ranking in social question-answering portals," in *Proc. 3rd Int. Workshop Search Mining User-Generated Contents*, 2011, pp. 19–26.

[28] Y. Cao, J. Xu, T.-Y. Liu, H. Li, Y. Huang, and H.-W. Hon, "Adapting ranking SVM to document retrieval," in *Proc. 29th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2006, pp. 186–193.

[29] X. Li, G. Cong, X.-L. Li, T.-A. N. Pham, and S. Krishnaswamy, "Rank-GeoFM: A ranking based geographical factorization method for point of interest recommendation," in *Proc. 38th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2015, pp. 433–442.

[30] J. Xu and H. Li, "AdaRank: A boosting algorithm for information retrieval," in *Proc. 30th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2007, pp. 391–398.

[31] X. Li, M. K. Ng, and Y. Ye, "MultiComm: Finding community structure in multi-dimensional networks," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 4, pp. 929–941, Apr. 2014.

[32] W. Wei, G. Cong, C. Miao, F. Zhu, and G. Li, "Learning to find topic experts in Twitter via different relations," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 7, pp. 1764–1778, Jul. 2016.

[33] W. Wei, B. Gao, T. Liu, T. Wang, G. Li, and H. Li, "A ranking approach on large-scale graph with multidimensional heterogeneous information," *IEEE Trans. Cybern.*, vol. 46, no. 4, pp. 930–944, Apr. 2016.

[34] X.-J. Wang, X. Tu, D. Feng, and L. Zhang, "Ranking community answers by modeling question-answer relationships via analogical reasoning," in *Proc. 32nd Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2009, pp. 179–186.

[35] P. Jurczyk and E. Agichtein, "Discovering authorities in question answer communities by using link analysis," in *Proc. 16th ACM Conf. Conf. Inf. Knowl. Manage.*, 2007, pp. 919–922.

[36] J. Zhang, M. S. Ackerman, and L. Adamic, "Expertise networks in online communities: Structure and algorithms," in *Proc. 16th Int. Conf. World Wide Web*, 2007, pp. 221–230.

[37] V. R. Carvalho, J. L. Elsas, W. W. Cohen, and J. G. Carbonell, "Suppressing outliers in pairwise preference ranking," in *Proc. 17th ACM Conf. Inf. Knowl. Manage.*, 2008, pp. 1487–1488.

[38] Z. Zheng, K. Chen, G. Sun, and H. Zha, "A regression framework for learning ranking functions using relative relevance judgments," in *Proc. 30th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2007, pp. 287–294.

[39] X. Wei, H. Huang, C. Lin, X. Xin, X. Mao, and S. Wang, "Re-ranking voting-based answers by discarding user behavior biases," in *Proc. 24th Int. Conf. Artif. Intell.*, 2015, pp. 2380–2386.

[40] T. Joachims, "Optimizing search engines using clickthrough data," in *Proc. 8th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2002, pp. 133–142.

[41] T. Joachims, L. Granka, B. Pan, H. Hembrooke, and G. Gay, "Accurately interpreting clickthrough data as implicit feedback," in *Proc. 28th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2005, pp. 154–161.

[42] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, pp. 27:1–27:27, 2011.

[43] L. Nie, Y. Zhao, X. Wang, J. Shen, and T. Chua, "Learning to recommend descriptive tags for questions in social forums," *ACM Trans. Inf. Syst.*, vol. 32, no. 1, 2014, Art. no. 5.

[44] Q. Le and T. Mikolov, "Distributed representations of sentences and documents," in *Proc. 31st Int. Conf. Mach. Learn.*, 2014, pp. 1188–1196.

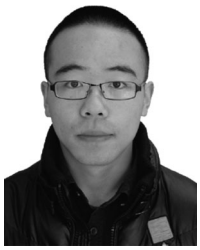
[45] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, 2003.

Liqiang Nie received the BE degree from the Xi'an Jiaotong University of China, Xi'an, in 2009 and the PhD degree from the National University of Singapore, in 2013. He is currently a professor in the School of Computer Science and Technology, Shandong University. Prior that, he was a research fellow in the School of Computing, National University of Singapore. His research interests include multimeida computing, information retrieval, and their applications in healthcare analytics. Various parts of his work have been published in top forums, such as ACM SIGIR, ACM MM, the *ACM Transactions on Information Systems*, and the *IEEE Transactions on Multimedia*. He has served as a reviewer for various journals and conferences.

Xiaochi Wei received the BE degree in computer science and technology from Liaoning University, Shenyang, in 2012. Currently, he is working toward the PhD degree in the School of Computer Science, Beijing Institute of Technology. His research interests include question answering, information retrieval, and natural language processing. He is a student member of the CCF and the ACM.

Dongxiang Zhang received the BSc degree from Fudan University, China, in 2006 and the PhD degree from the National University of Singapore, in 2012. He is a professor in the School of Computer Science and Engineering, University of Electronic Science and Technology of China. He worked as a research fellow with NExT Research Center, Singapore, from 2012 to 2014 and was promoted as a senior research fellow in 2015. His research interests include spatial databases, cloud computing, and big data analytics.

Xiang Wang received the BE degree in computer science and engineering from Baihang University, Beijing, in 2014. He is currently working toward the PhD degree in the School of Computing, National University of Singapore. His research interests include information retrieval, social media analysis, and grouping discovery and profiling from social media. He has served as a reviewer for various conferences, such as MMM and WebScience.



Zhipeng Gao is currently working toward the master's degree in the School of Software Engineering, University of Science and Technology of China. He worked as an intern student in the School of Computing, National University of Singapore, from August 2015 to June 2016. During his internship, he worked with Dr. Liqiang Nie and Mr. Xiaochi Wei. His research interests include text data mining and question answering.

Yi Yang received the PhD degree in computer science from Zhejiang University, Hangzhou, China, in 2010. He is currently an associate professor with the University of Technology Sydney, Australia. He was a post-doctoral researcher in the School of Computer Science, Carnegie Mellon University, Pittsburgh, Pennsylvania. His current research interests include machine learning and its applications to multimedia content analysis and computer vision, such as multimedia indexing and retrieval, surveillance video analysis, and video content understanding.

▷ **For more information on this or any other computing topic, please visit our Digital Library at** www.computer.org/publications/dlib.