Reusing Keywords for Fine-grained Representations and Matchings

Li Chong
Renmin University of China
Beijing, China
chongli@ruc.edu.cn

Denghao Ma
Meituan
Beijing, China
madenghao5@gmail.com

Yueguo Chen
Renmin University of China
Beijing, China
chenyueguo@ruc.edu.cn

ABSTRACT

Question retrieval aims to find the semantically equivalent questions for a new question, suffering from a key challenge—lexical gap. Previous solutions mainly focus on the translation model, topic model and deep learning techniques. Distinct from the previous solutions, we propose new insights of reusing important keywords to construct fine-grained semantic representations of questions and then fine-grained matchings for estimating the semantic similarity of two questions. Accordingly, we design a fine-grained matching network by reusing the important keywords. In the network, two cascaded units are proposed: (i) fine-grained representation unit, which uses multi-level keyword sets to represent question semantics of different granularity; (ii) fine-grained matching unit, which first generates multiple comparable representation pairs for two questions, i.e., keyword set pairs, and then matches the two questions from multiple granularities and multiple views by using the comparable representation pairs, i.e., from global matching to local matching and from lexical matching to semantic matching. To get the multi-level keyword sets of a question, we propose a cross-task weakly supervised extraction model that applies question-question labeled signals from the training set of question retrieval to supervise the keyword extraction process. To construct the comparable keyword set pairs, we design a pattern-based assignment method to construct the comparable keyword set pairs from the multi-level keyword sets of two questions. We conduct extensive experiments on three public datasets and the experimental results show that our proposed model outperforms the state-of-the-art solutions.

CCS CONCEPTS

• Computer systems organization → Embedded systems; Redundancy; Robotics; • Networks → Network reliability.

KEYWORDS

Question answering, Question retrieval, Semantic representation

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

Conference acronym ‘XX, June 03–05, 2018, Woodstock, NY © 2018 Association for Computing Machinery.
ACM ISBN 978-1-4503-XXXX-X/18/06.. $15.00
https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Question retrieval is to retrieve semantically equivalent questions from an archived repository with large quantities of questions. It has been verified very important to both community-based question answering (CQA) [14], e.g., Yahoo! Answers, Quora and Baidu Knows, and domain-specific question answering, e.g., Apple Siri and Microsoft Azure Bot. Although many researchers from the communities of academia and industry have paid much attention to the task, it still suffers from a key challenge, i.e., the lexical gap. The lexical gap contains two aspects: 1) textually distinct yet semantically equivalent; 2) textually similar yet semantically distinct. For example in Figure 1, given three questions $q_1$ = "how to keep the mobile phone cool", $d_1$ = "stop my iphone from overheating" and $d_2$ = "how to keep the mobile phone fast", $q_1$ and $d_1$ are textually distinct yet semantically equivalent, while $q$ and $d_2$ are textually similar yet semantically distinct. The lexical gap hinders the effective retrieval of semantically equivalent questions from the archives to a user’s query (a question or some keywords).

To address the challenge of textually distinct yet semantically equivalent, many solutions have been proposed based on the translation model [13], topic model [8], and deep learning techniques [19]. The solutions based on translation model [13, 30] learn translation probabilities for both words and phrases over some parallel corpora, and then use the translation probabilities to estimate the semantic similarity of two questions. The solutions based on topic model [1, 8, 15] construct the latent topic distributions for questions and estimate the semantic similarity of the two questions based on their topic distributions. Deep learning solutions mainly focus on 1) modelling the semantic relations of two questions by different neural network architectures and then estimating their similarity based on the relations [16, 20, 28]; 2) constructing good semantic representations for questions, such as BERT [5] and ERNIE [22]. These solutions model the semantics of a question by using a global representation, i.e., a bag of words, topic distribution and embedding, and globally match two questions. Just because of the "global representation", questions with similar text yet distinct semantics have similar topic distribution representation and embedding representation; Because of the "globally match", the above solutions hardly distinguish the subtle differences of important keywords of two questions and thus can not effectively address the challenge of textually similar yet semantically distinct.
To address the two aspects of the lexical gap, this paper attempts to find answers for the following two research questions: 1) How to represent questions? 2) How to match two questions?

**Insight one: multi-level keyword sets.** We propose new insights of using important keywords to construct fine-grained semantic representations of questions and then fine-grained matchings. For example in Figure 1, given two questions $q_1$="how to keep the mobile phone cool" and $d_1$="stop my iphone from overheating", their corresponding keywords can be extracted and assigned different importance scores; Based on the importance scores, the multi-level keyword sets can be generated, i.e., the question $q_1$ has 7 keyword sets and $d_1$ has 5 keyword sets; For $q_1$, $KS_1(q_1)$ represents the finest semantic representation of $q_1$, and $KS_1(q_1)$ represents the coarsest one of $q_1$. The multi-level keyword sets can model the question semantics of various granularity, i.e., from "finest" to "coarsest". The finest keyword set can be used for modelling the global semantics of a question and then addressing the challenge of textually distinct yet semantically equivalent. The other sets are using the important keywords so that the subtle differences of important keywords of two questions can be distinguished and then the challenge of textually similar yet semantically distinct can be addressed by the fine-grained matchings. For example in Figure 1, for the question $q_1$ and $d_2$, the matchings between $KS_1(q_1)$ and $KS_1(d_2)$ as well as $KS_2(q_1)$ and $KS_2(d_2)$ can effectively identify the semantic distinction between $q_1$ and $d_2$.

**Insight two: comparable keyword set pairs.** Given two questions and their multi-level keyword sets, how to match them? For example in Figure 1, the question $q_1$ and $d_2$ have 5 and 7 keyword sets respectively. We match $KS_1(q_1)$ to $KS_1(d_1)$, $KS_2(q_1)$ to $KS_1(d_1)$, $KS_1(q_1)$ to $KS_2(d_1)$, $KS_2(q_1)$ to $KS_2(d_1)$, $KS_2(q_1)$ to $KS_3(d_1)$, $KS_3(q_1)$ to $KS_3(d_1)$, because each pair is at the same or similar semantic level and is comparable. Reversely, $KS_1(q_2)$ and $KS_1(d_2)$ as well as $KS_2(q_2)$ and $KS_2(d_2)$ are not comparable because they are at different semantic level. The matchings on comparable keyword set pairs can benefit the similarity estimation of two questions, while that on the incomparable keyword set pairs will hurt the similarity estimation. Therefore, to effectively estimate the similarity of two questions, we need to construct the comparable keyword set pairs for the two questions.

**Solution: fine-grained matching network.** Based on the insights, we propose the fine-grained matching network with two cascaded units to estimate the similarity of two questions.

**Fine-grained Representation Unit.** According to the insight one, the fine-grained representation unit constructs the multi-level keyword sets of a question for modelling its fine-grained semantics. Because of the deficiency of keyword annotations, we are the first to use the question-question labeled signals from the training set of question retrieval to supervise the keyword extraction (i.e., learn the keyword importance), and develop a cross-task weakly supervised extraction model. In the model, we integrate deep semantic representations, lexical information, part-of-speech and statistical features to estimate the importance of keywords in a question. Based on the importance, the multi-level keyword sets are generated by iteratively removing one keyword with the lowest importance.

**Fine-grained Matching Unit.** According to the insight two, we design a pattern-based assignment method to construct the comparable keyword set pairs from the multi-level keyword sets of two questions. The fine-grained matching unit estimates the semantic similarity of two questions by matching their comparable keyword set pairs. First, it matches two questions from the global matching to local matching by using the comparable pairs of different granularity. Second, it matches every comparable pair from both the semantic matching and lexical matching perspectives. For the semantic matching, we develop two types of methods, i.e., MLP-based matching and attention-based matching; For the lexical matching, we use some widely used textual matching models such as BM25 and the Jaccard similarity. Third, it aggregates these matchings of multiple comparable pairs by learning their weights, and outputs an overall score as the semantic similarity of two questions.

Our contributions are concluded as follows:
- We propose new insights of using important keywords to construct the fine-grained representations and matchings, and design
the fine-grained matching network to estimate the similarity of two questions from multiple granularities and multiple views.

* We are the first to use question-question labeled signals to supervise the keyword extraction process and develop the cross-task weakly supervised extraction model.
* We propose a pattern-based assignment method to construct the comparable keyword set pairs.
* We conduct extensive experiments on three public datasets and validate the effectiveness of the fine-grained matching network.

2 RELATED WORK

Community-based question answering (CQA) portals have been popular platforms in which users may ask their questions and share their answers. Because of the opening and sharing mechanism, CQA portals have accumulated a large quantity of questions and answers, which has made CQA portals valuable resources. As the quantity increases, users’ questions may be repeated or closely related to previously archived questions which may be replied with some answers. Therefore, CQA portals first retrieve semantically equivalent questions (i.e. question retrieval) from the archived repository and then identify the high-quality answers (i.e. answer selection) of these equivalent questions. So question retrieval is an important task in CQA portals.

The question retrieval task suffers from the lexical gap challenge, and many solutions have been proposed to address it. We survey these solutions and group them into three types:

**Solutions based on translation model.** These solutions define the similarity of two questions as the translation probability from one to the other [7, 27]. The study [27] proposes a retrieval model consisting of a translation-based language model for the question part and a query likelihood approach for the answer part. The proposed model incorporates word-to-word translation probabilities learned from different parallel corpora. The study [30] proposes a phrase-based translation model for question retrieval.

**Solutions based on topic model.** These solutions [1, 8, 29] define the similarity in the latent topic space. The work [1] proposes a topic model that incorporates category information into the process of discovering latent topics. The work [8] proposes a Question-Answer Topic Model (QATM) that learns the latent topics from the question-answer pairs, by assuming that a question and its paired answer share the same topic distribution. In the work [29], a topic-based language model is proposed to match questions not only at a term level but also at a topic level.

**Deep learning solutions.** The deep learning techniques have been widely applied in question retrieval, and achieve better performance than traditional solutions. The work [20] designs a convolutional neural network that learns an optimal representation of question pairs and a similarity function to relate them in a supervised way from the available training data. The work [25] uses a bidirectional LSTM to generate multiple positional sentence representations for each question. In the work [4], the Siamese CNN is proposed to estimate the semantic similarity of two questions.

Recently, the pre-training language models have demonstrated strong performance on text representations in many NLP tasks, such as BERT [5], Sentence-BERT [18] and ERNIE [22]. Meanwhile, many models are proposed based on various pre-trained language models to estimate the similarity of two questions. The work [19] presents a BERT-based FAQ retrieval system which uses BERT to estimate the correlation between a question and an answer. The work [12] develops a fully unsupervised method which uses question answer pairs to train two BERT models and uses the BERT models to match user queries to answers and questions, respectively. The work [15] proposes a topic-informed BERT-based architecture to estimate the similarity of the two texts. In work [17], the medical entity information is learned via ERNIE and incorporated into some similarity estimation models.

3 FINE-GRAINED MATCHING NETWORK

In this section, we first formulate the question retrieval problem, and subsequently introduce our proposed fine-grained matching network with two cascaded units, i.e., fine-grained representation unit and fine-grained matching unit.

3.1 Problem and The Overall Solution

Question retrieval is to find semantically equivalent questions from the archived repository for a new question issued by a user. It mainly consists of two stages: 1) the recalling stage retrieves a subset of question candidates from the whole question repository. At this stage, many solutions have been proposed based on text matching and semantic matching; 2) The ranking stage sorts the candidates to generate the final results by using some fine-grained matching methods. In this paper, we focus on the ranking stage. Similar to many studies [10, 26], we assume that the question candidates exist. Given a new question \( q \) and the candidates \( D \), we formulate the semantic similarity of each \( d \in D \) to \( q \) as follows:

\[
\text{sim}(q, d) = f(q, d, \Phi)
\]  

(1)

where the function \( f \) takes \( q, d \) and a set of parameters \( \Phi \) as input and outputs a score ranging from 0 to 1 as the similarity between \( q \) and \( d \). According to the scores, the candidates in \( D \) can be ranked in descending order. To design an effective function \( f \), we need to answer two research problems—how to represent questions and how to match the two questions.

As mentioned in Section 1, to answer the above two questions, we propose the fine-grained matching network with two cascaded units. The fine-grained representation unit uses the multi-level keyword sets to represent the fine-grained semantics of a question. The fine-grained matching network first constructs multiple comparable pairs of keyword sets for two questions, and then matches two questions from multiple granularities and multiple perspectives by using comparable keyword set pairs, i.e., from global matching to local matching and from lexical matching to semantic matching. The two units are simultaneously optimized by feedforward and backward propagation. Given a question \( q \) and a candidate question \( d \), we formulate the network as follows:

\[
f(q, d, \Phi) = \text{MLP}(w_2E(S(q, d)) + b_2)
\]  

(2)

where \( S(q, d) \) denotes the multiple comparable pairs of keyword sets and \( E(S(q, d)) \) is the multi-view matching representation over \( S(q, d) \), which are constructed in Section 3.3. The \( \text{MLP} \) with parameters \( w_2 \) and \( b_2 \) is an \( m \times n \times 1 \) multi-layer perceptron where the activation functions of the first two layers are ReLU [3] and that of the third layer is Sigmoid [21].
3.2 Fine-grained Representation Unit

As motivated in Section 1, to construct the multi-level keyword sets of a question, we need to extract and score keywords for a question. However, it is hard to apply a supervised model to extract and score keywords due to the lack of keyword annotations. Simply applying unsupervised models suffers from the precision of keyword extraction, which will be verified in the experiments. These motivate us to use question-question labeled signals to supervise the keyword extraction process, leading to a cross-task weakly supervised extraction strategy.

Feasibility of the cross-task weakly supervised strategy: On the one hand, accurate keyword extraction can benefit the similarity estimation. For example in Figure 1, given the user’s question “how to keep the mobile phone cool?” and the relevant archived question “stop my iphone from overheating”, if the keyword extracting can effectively extract their corresponding keywords, e.g., “how”, “to”, “keep”, “phone”. The fined-grained matchings will be able to effectively estimate their similarity. Conversely, inaccurate keyword extraction will hurt the similarity estimation. On the other hand, keywords with importance scores and fine-grained matching scores are propagated to the loss function, and the loss can be backward propagated to keyword extraction component. Therefore, we need to estimate the importance of a keyword candidates. For example, given a Chinese question “我的苹果手机会发热”，we can use some commonly used segmentation tools, e.g., pkuseg1, to segment each question, and take these segments as keyword candidates. For example, given a Chinese question “为什么我的苹果手机会发热”, the segmentations are “为什么/why”, “我的/my”, “苹果手机/iphone”, “会/does”, “发热/overheat”. The importance of keywords in a question can be different to represent the meaning of the question. For the above example, the keyword candidate “苹果手机/iphone” and “发热/overheat” are more important to represent the semantics of the question than other ones. Therefore, we need to estimate the importance of a keyword set to q, and develop the importance estimation network as follows:

\[ w(k_i, q) = \text{ImNN}(k_i, q) \quad (3) \]

where \( \text{ImNN} \) is a neural network that takes \( k_i \) and \( q \) as input, and outputs an importance score ranging from 0 to 1. We design the network \( \text{ImNN} \) with three subnets: 1) \( \text{EmbNN} \) learns the embedding representations for \( k_i \) and \( q \), and constructs their co-action embedding representations; 2) \( \text{EFNN} \) constructs the external features of \( k_i \); 3) \( \text{JINN} \) processes the joint information from the previous two subnets and outputs the importance score.

In \( \text{EmbNN} \), we use BERT [5] to generate the embedding of each candidate keyword. For Chinese questions, we first use BERT to generate the embedding \( E_w \) for each character \( w \in k_i \), and then use the embedding pooling of characters in \( k_i \) as the embedding of \( k_i \), i.e., \( E_{k_i} = \sum_{w \in k_i} E_w / |k_i| \) where \( |k_i| \) is the textual length of \( k_i \). For the question \( q \), the embedding \( E_{[CLS]} \) is taken as the semantic embedding of \( q \) (denoted as \( E_{CLS} \)). Based on \( E_{k_i} \) and \( E_{CLS} \), we formulate \( \text{EmbNN} \) as follows:

\[ \overline{emb}_f = \text{MLP}(w_f[E_{CLS} \odot E_{k_i} \odot \text{CoA}(E_{CLS}, E_{k_i})]) + b_f \quad (4) \]

\[ \text{CoA}(E_{CLS}, E_{k_i}) = \sigma_\theta(w_o[E_{CLS} \odot E_{k_i} \odot |E_{CLS} - E_{k_i}|] + b_o) \quad (5) \]

where \( \odot \) is the concatenating operation and \(| \cdot | \) is a sign of computing the absolute value. The \( \sigma_\theta \) is the ReLU activation function and MLP is a multi-layer perceptron with three layers whose activation functions are ReLU. The \( w_o \) and \( b_o \) are the weight matrix and the bias vector.

The subnet \( \text{EFNN} \) is to construct the external features for the keyword \( k_i \). The inverse document frequency (IDF)—the inverse fraction of the documents that contain the word, measures the quantity of information the word provides. To improve the generalization of our model, we split all IDF values into \( K \) buckets and initialize each bucket into a learnable embedding with 64 dimensions. In our experiments, \( K = 100 \) by default. Therefore, the IDF value of \( k_i \) is represented by an embedding, i.e., \( \text{EIDF}_{k_i} \). The part of speech (POS) feature has been verified important to many NLP tasks [9, 23]. We use the pyltp tool2 to mark the part of speech of each keyword candidate in Chinese sentences, and the Stanford POS tagger3 for English sentences. We train each POS into a learnable embedding, and denote the POS embedding of \( k_i \) as \( \text{EPOS}_{k_i} \). Besides, we construct the stop-word feature for each word with 0 or 1 value, by using some stop-word dictionaries4. We denote the stop-word feature of \( k_i \) as \( \text{ESW}_{k_i} \). Based on the \( \text{EIDF}_{k_i} \), \( \text{EPOS}_{k_i} \) and \( \text{ESW}_{k_i} \) features, we formulate \( \text{EFNN} \) as follows:

\[ \overline{ef}_f = \text{MLP}(w_f[\text{EIDF}_{k_i} \odot \text{EPOS}_{k_i} \odot \text{ESW}_{k_i}] + b_f) \quad (6) \]

where MLP is a multi-layer perceptron with three layers whose activation functions are ReLU.

In the \( \text{JINN} \) subnet, we first concatenate the embedding-based features \( \overline{emb}_f \) and the external features \( \overline{ef}_f \), and then use an MLP with three layers to estimate the importance of \( k_i \) to \( q \). Based on the \( \text{EmbNN} \), \( \text{EFNN} \) and \( \text{JINN} \) subnets, we formulate the neural network \( \text{ImNN} \) in Equation 3 as follows:

\[ \text{ImNN}(k_i, q) = \text{MLP}(w_f[\overline{emb}_f \odot \overline{ef}_f] + b_f) \quad (7) \]

where MLP is a multi-layer perceptron with three layers. The activation functions of the first two layers are ReLU, and that of the last layer is Sigmoid.

According to the keywords and their importance, we remove one keyword with the lowest importance from the keyword set one by one, and each removal generates a keyword set. For example in Figure 1, given question \( q \) = “how to keep the mobile phone cool” and its keywords [“how”: 0.3, “to”: 0.1, “keep”: 0.5, “the”: 0.05, “mobile”: 0.4, “phone”: 0.85, “cool”: 0.8] and [“stop”: 0.3, “my”: 0.1, “iphone”: 0.85, “from”: 0.05, “overheating”: 0.8], the fined-grained matchings will be able to effectively estimate their similarity. Conversely, inaccurate keyword extraction will hurt the similarity estimation. On the other hand, keywords with importance scores and fine-grained matching scores are propagated to the loss function, and the loss can be backward propagated to keyword extraction component. Therefore, we need to estimate the importance of a keyword candidates. For example, given a Chinese question “为什么我的苹果手机会发热”, we can use some commonly used segmentation tools, e.g., pkuseg1, to segment each question, and take these segments as keyword candidates. For example, given a Chinese question “为什么我的苹果手机会发热”, the segmentations are “为什么/why”, “我的/my”, “苹果手机/iphone”, “会/does”, “发热/overheat”. The importance of keywords in a question can be different to represent the meaning of the question. For the above example, the keyword candidate “苹果手机/iphone” and “发热/overheat” are more important to represent the semantics of the question than other ones. Therefore, we need to estimate the importance of a keyword \( k_i \) to \( q \), and develop the importance estimation network as follows:

\[ w(k_i, q) = \text{ImNN}(k_i, q) \quad (3) \]

where \( \text{ImNN} \) is a neural network that takes \( k_i \) and \( q \) as input, and outputs an importance score ranging from 0 to 1. We design the network \( \text{ImNN} \) with three subnets: 1) \( \text{EmbNN} \) learns the embedding representations for \( k_i \) and \( q \), and constructs their co-action embedding representations; 2) \( \text{EFNN} \) constructs the external features of \( k_i \); 3) \( \text{JINN} \) processes the joint information from the previous two subnets and outputs the importance score.

In \( \text{EmbNN} \), we use BERT [5] to generate the embedding of each candidate keyword. For Chinese questions, we first use BERT to generate the embedding \( E_w \) for each character \( w \in k_i \), and then

---

1. https://github.com/pkuclu/pkuseg-python
2. https://github.com/goto456/stopwords; https://gist.github.com/sebleier/554280
3. https://nlp.stanford.edu/software/tagger.shtml
4. https://github.com/HIT-SCIR/pyltp
5. https://github.com/gotos56/stanfordnlp; https://gist.github.com/sebleier/554280
3.3 Fine-grained Matching Unit

3.3.1 Comparable Keyword Set Pairs. As motivated in Section 1, the matchings should be performed on the comparable keyword set pairs. So we formally define a comparable pair as follows: Comparable keyword set pair: given two questions $q$ and $d$, and their multi-level keyword sets $KS(q)$ and $KS(d)$, a comparable keyword set pair is defined as $S_i = KS_m(q), KS_n(d)$ where $KS_m(q) \in KS(q)$ and $KS_n(d) \in KS(d)$ are two keyword sets that are at the same or similar semantic level.

We develop the pattern-based assignment method to construct the comparable pair of keyword sets. First, we compute the ratio that each keyword set covers the keywords in the highest level keyword set. For example in Figure 1, given $q$ and the keyword sets $KS(q) = \{KS_1(q), \ldots, KS_5(q)\}$, the ratio of $KS_1(q)$ is $|KS_1(q)|/|KS_1(q)| = 1$, that of $KS_2(q)$ is $|KS_2(q)|/|KS_1(q)| = 0.857$, and that of $KS_3(q)$ is $|KS_3(q)|/|KS_1(q)| = 0.714$. Second, given two questions and their corresponding keyword sets and ratios, we extract keyword sets with the same or similar ratios as comparable pairs of keyword sets. Specifically, we use patterns to extract comparable keyword sets: given a user question $q$ and a keyword set $KS_m(q)$ with ratio $r_m(q)$, 1) if $KS_m(q)$ contains multiple keywords (i.e., $|KS_m(q)| > 1$) and there is a keyword set $KS_n(d) \in KS(d)$ with the ratio $r_m(q)$, we record $KS_m(q)$ and $KS_n(d)$ as a comparable pair; 2) if $|KS_m(q)| > 1$ and there is no keyword set in $KS(d)$ with the ratio $r_m(q)$, we extract the keyword set $KS_n(d)$ whose ratio is larger than $r_m(q)$ and is the closest to $r_m(q)$, and record $KS_n(d)$ and $KS_m(q)$ as a comparable pair; 3) if $|KS_m(q)| == 1$, we use $KS_n(d)$ with only one keyword to pair $KS_m(q)$; 4) otherwise, we don’t use $KS_m(q)$ to construct comparable pairs. By the pattern-based assignment method, a set of comparable pairs are constructed for the question $q$ and $d$. We denote the comparable pairs as $S(q, d) = \{\cdots, <KS_m(q), KS_n(d)>, \cdots\}$.

3.3.2 Multi-view Matching. According to Equation 2, we propose the multi-view matching over $S(q, d)$ to construct $E(S(q, d))$:

$$E(S(q, d)) = \frac{\sum_{i=0}^{n} w(S_i(q, d)) m(o(S_i(q, d)))}{\sum_{i=0}^{n} w(S_i(q, d))}$$

(8)

where $m(o(S_i(q, d)))$ is the multi-view matching on the pair $S_i(q, d)$. The $w(S_i(q, d))$ is the weight of $S_i(q, d)$, because the matchings on the different comparable pairs have different importance.

The pair $S_i(q, d)$ contains two keyword sets $KS_m(q)$ and $KS_n(d)$. If $KS_m(q)$ and $KS_n(d)$ are closer to the semantics of $q$ and $d$, the weight should be higher. In the keyword set, each keyword $k_i \in KS_m(q)$ or $k_j \in KS_n(d)$ is assigned an importance. Based on the importance, we estimate the weight $w(S_i(q, d))$ as follows:

$$w(S_i(q, d)) = \frac{\sum_{k_i \in KS_m(q)} w(k_i, q)}{|q|} \cdot \frac{\sum_{k_j \in KS_n(d)} w(k_j, d)}{|d|}$$

(9)

where $w(k_i, q)$ is estimated by using Equation 7. The $|q|$ and $|d|$ are lengths of the question $q$ and $d$, respectively.

To effectively match the keyword set pair $KS_m(q)$ and $KS_n(d)$ ($S_i(q, d)$), we design a multi-view matching model from both semantic matching and lexical matching perspectives. The multi-view matching model is formulated as follows:

$$m_{o}(S_i(q, d)) = m_{mlp}(KS_m(q), KS_n(d)) \oplus m_{att}(KS_m(q), KS_n(d)) \oplus m_{lm}(KS_m(q), KS_n(d))$$

(10)

where $m_{mlp}$ is the MLP-based matching, $m_{att}$ is the attention-based matching, and $m_{lm}$ matches them from the lexical perspective. We design the MLP-based matching $m_{mlp}$ as follows:

$$m_{mlp}(KS_m(q), KS_n(d)) = \sigma(w_m[KS_m(q) @ KS_n(d)] + CoA(KS_m(q), KS_n(d))] + bm)$$

(11)

where $KS_m(q)$ and $KS_n(d)$ are the semantic representations of $KS_m(q)$ and $KS_n(d)$. The CoA($KS_m(q), KS_n(d)$) can be estimated by using Equation 5. We estimate $KS_m(q)$ as follows:

$$KS_m(q) = \sum_{k_i \in KS_m(q)} w(k_i, q) E_i$$

(12)

where $E_i$ is the embedding representation of $k_i$, which can be learned by some pre-trained language models. The $KS_n(d)$ is constructed in the same way.

The MLP-based matching ignores the dependence of keywords. So we design the attention-based matching to incorporate the dependence into the matching representation. Specifically, we concatenate the words in $KS_m(q)$ and $KS_n(d)$ separately, and denote the concatenated texts as $d$ and $q$. Secondly, we concatenate $q$ and $d$ by using [CLS] to denote the start position and [SEP] to differentiate them. We initialize the concatenated text with a list of embedding vectors $L$ and use the multi-head scaled dot-product attention [24] to model the dependence among keywords:

$$head_{1} = softmax\left(\frac{LW^q_i Q_i^T + LW^d_i K_i T}{\sqrt{d_k}}\right)$$

(13)

$$head_{2} = [head_{1}, \cdots, head_{n}] W_o$$

(14)

where $W_i^q, W_i^d, W_i$ and $W_o$ are learnable weights. We use the position-wise feed forward network (FFN) and layer normalization (LNorm) to further refine $head_{1}$:

$$m_{att}(KS_m(q), KS_n(d)) = LNorm(head_{1} + FFN(head_{1}))$$

(15)

where $m_{att}$ is the attention-based matching representation.

Different from $m_{mlp}$ and $m_{att}$, $m_{lm}$ matches $KS_m(q)$ and $KS_n(d)$ from the lexical perspective. It uses multiple solutions to match $KS_m(q)$ and $KS_n(d)$. The function $m_{lm}$ is formulated as follows:

$$m_{lm}(KS_m(q), KS_n(d)) = BM25 \odot Jaccard \odot wlm(KS_m(q), KS_n(d))$$

(16)

$$wlm(KS_m(q), KS_n(d)) = \sum_{k_i \in KS_m(q)} \sum_{k_j \in KS_n(d)} w(k_i, q) w(k_j, d) a(k_i, k_j)$$

(17)

where BM25 is the Okapi BM25 matching function, and $Jaccard$ is the Jaccard similarity coefficient. The $a(k_i, k_j)$ equals 1 if $k_i = k_j$, otherwise $a(k_i, k_j)$ equals 0. The $w(k_i, q)$ is the importance score of $k_i$ to $q$, and it is estimated by using Equation 7.

5https://en.wikipedia.org/wiki/Okapi_BM25
6https://en.wikipedia.org/wiki/Jaccard_index
Table 1: Statistical information of datasets.

| Dataset   | Type    | Train | Dev  | Test |
|-----------|---------|-------|------|------|
| BankQ     | positive| 50000 | 5000 | 5000 |
|           | negative| 50000 | 5000 | 5000 |
| Quora     | positive| 139306| 5000 | 5000 |
|           | negative| 245042| 5000 | 5000 |
| LCQMC     | positive| 138574| 4402 | 6250 |
|           | negative| 100192| 4400 | 6250 |

3.3.3 Training and Inference. At the training stage, we use the Log-likelihood loss to optimize the fine-grained matching network and present the loss function as follows:

\[
L = -\frac{1}{|C|} \sum_{<q,d,y> \in C} \{y \log(sim(q,d)) + (1-y) \log(1 - sim(q,d))\}
\]  

(18)

where \(C\) is a set of training examples, and \(|C|\) is the size of \(C\). The \(y\) is the ground truth label of \(q\) and \(d\), and \(y = 1\) denotes that \(d\) is semantically equivalent to \(q\). The \(sim(q,d)\) is the similarity between \(q\) and \(d\), which is estimated by using Equation 1. At the inference stage, only if \(sim(q,d) > 0.5\), \(d\) is identified semantically equivalent to \(q\).

4 EXPERIMENTS

4.1 Experimental Setup

4.1.1 Datasets Description. We conduct experiments on three public datasets, show the statistical information of the datasets in Table 1, and introduce them as follows:

- BankQ [2]: It is the largest dataset of question retrieval in the financial domain and sampled from the session logs of an online bank custom service system.
- Quora [3]: The dataset is sampled from the American Knowledge Q&A website Quora.com. Each question pair is labeled with a binary value indicating whether the two questions are similar.
- LCQMC [10]: It is a large-scale Chinese question matching corpus and sampled from the largest online Chinese question answering platform, i.e., Baidu Knows.

4.1.2 Comparison Solutions. We select some state-of-the-art solutions as baselines to verify the effectiveness of our model:

- BiMPM [26]: It first encodes the question \(q\) and \(d\) with a BiLSTM encoder, and then matches the two questions by using multiple matching methods, i.e., full-matching, max-pooling-matching, attentive-matching, and max-attentive-matching.
- RE2 [28]: It is a fast and strong neural architecture for general-purpose text matching. It integrates the previously aligned features (Residual vectors), original point-wise features, and contextual features as input of the inter-sequence alignment.
- BERT [5]: It is designed to pre-training deep bidirectional representations and achieves new state-of-the-art results on many natural language processing tasks including question retrieval.
- ERNIE [22]: It is a continual pre-training framework to effectively capture the lexical, syntactic, and semantic information from training corpora. It uses the constant multi-task learning strategy to learn incrementally some pre-training tasks.

- Sentence-BERT [18]: Using the framework of the twin network model, it inputs different sentences into the parameter shared BERT model, and outputs the embedding vector for each sentence.
- tBERT [15]: It uses topic models, such as LDA and GSDMM, to improve BERT for pairwise semantic similarity detection.

We denote our proposed model as FRM.

4.1.3 Performance metrics. Similar to the studies [15, 22, 28], we use accuracy and AUC metrics to measure the effectiveness of all models, and introduce them as follows:

- Accuracy (Acc.): It is a widely used metric for evaluating classification models, and measures the fraction of correct predictions. Based on the positives and negatives, Acc. can be calculated as Acc. = \(\frac{TP + TN}{TN + FP + FN + TP}\), where \(TP =\) true positives, \(TN =\) true negatives, \(FP =\) false positives, and \(FN =\) false negatives.
- Area Under Curve (AUC.) [6]: AUC. measures the two-dimensional area under the ROC curve. It represents the probability that a positive example is positioned in front of a random negative example.

4.1.4 Reproducibility. The parameters of all models are assigned the default values for fair comparisons. Specifically, the batch sizes of models on the BankQ and LCQMC datasets are 32 and that of models on Quora datasets is 64. All models are developed in Python 3.8 and Pytorch 1.10 development environment. We set the learning rate to \(2e - 5\) and use the warm-up learning rate method. We apply dropout before every fully-connected layer of all models and set the dropout probability as 0.3. All models are deployed on the same computing services so that they have the same running conditions. In models, the parameters are optimized by the AdamW optimizer [11]. For fair comparisons, we perform every solution five times, and choose the best model to evaluate on the test set.

4.2 Experimental Results

A summary of the performance comparison over three datasets is displayed in Table 2, 3 and 4. According to the results, it can be
seen that 1) the performance achieved by the baselines BERT and ERNIE is higher than that achieved by the other baseline models. This verifies that BERT and ERNIE are very strong baseline models. Compare to BIMPM and RE2, BERT and ERNIE are pre-training representation models that benefit from the high-quality pre-training representations; 2) The performance achieved by FRM is better than that achieved by the best baseline models, i.e., BERT and ERNIE. Compared to BERT and ERNIE, FRM benefits from fine-grained representations and fine-grained matchings. The performance improvements verify the effectiveness of FRM.

To further verify the effectiveness of our model and the better baselines in addressing the lexical gap challenge (i.e., textually distinct yet semantically equivalent and textually similar yet semantically distinct), we first classify the test samples into the hard, easy and middle types; Second, we investigate the Acc. metrics of models on the three types of samples, and report the results in Figure 2. About the classification of samples, we group the positive samples of textually distinct yet semantically equivalent (i.e., the Jaccard similarity is in $[0.7, 1.0]$) into the hard type; The positive samples of textually similar and semantically equivalent (i.e., the Jaccard similarity is in $[0.7, 1.0]$) and the negative samples of textually distinct and semantically distinct (i.e., the Jaccard similarity is in $[0.0, 0.2]$) are grouped into the easy type; The remained samples are grouped into the middle type. In Figure 2, it can be seen that 1) the Acc. values of all models on easy samples are higher than those on the middle samples, and the Acc. values of all models on middle samples are higher than those on the hard samples; 2) The FRM model performs better than BERT and ERNIE on the three types of samples. The advantage of FRM on the hard samples is more significant than that on easy and middle samples. These illustrate that FRM can better address the lexical gap challenge than BERT and ERNIE.

We investigate the effect of question lengths on the performance of FRM, BERT and ERNIE and show the results in Figure 3. It can be seen that 1) the Acc. values are lower when the question length is smaller or larger. This is because questions with few words can not provide rich semantics, and the models hardly capture the semantic focus of questions with too many words; 2) Overall, FRM performs better than BERT and ERNIE. This illustrates that the advantage of FRM is not affected by the question lengths, even though the construction of multi-level keyword sets depends on the question words.

The fine-grained matching unit matches two questions from the perspectives of both semantic matching and lexical matching. To achieve semantic matching, we develop the MLP-based matching function $m_{\text{mlp}}$ (see Equation 11) and the attention-based matching function $m_{\text{att}}$ (see Equation 15); To achieve lexical matching, we develop the matching function $m_{\text{lm}}$ (see Equation 16). To verify the effectiveness of the three matching functions and investigate their

Table 4: Performance comparison over Quora.

| Model     | Acc.(%) | AUC.(%) |
|-----------|---------|---------|
| BiMPM     | 85.82   | 93.46   |
| RE2       | 89.20   | 95.56   |
| BERT      | 91.07   | 96.96   |
| ERNIE     | 91.08   | 96.84   |
| Sentence-BERT | 88.69     | 94.84   |
| tBERT     | 89.31   | 95.69   |
| FRM       | 91.64   | 97.01   |
Table 5: The effectiveness of fine-grained matching unit.

| Model         | Acc.(%) | AUC.(%) |
|---------------|---------|---------|
| FRM-mlp       | 83.64   | 92.03   |
| FRM-att       | 83.93   | 91.58   |
| FRM-lm        | 84.71   | 91.79   |
| BERT+ef       | 84.27   | 92.05   |
| BERT+BM25     | 84.53   | 91.98   |
| BERT          | 84.06   | 89.34   |
| FRM           | 85.16   | 92.43   |

Table 6: The effectiveness of fine-grained representation unit.

| Model         | Acc.(%) | AUC.(%) |
|---------------|---------|---------|
| FRM-ef        | 84.38   | 92.20   |
| FRM-dsr       | 84.26   | 92.29   |
| FRM-lm        | 84.12   | 92.19   |
| FRM-TFIDF     | 83.91   | 92.18   |
| FRM-one level | 84.48   | 91.12   |
| FRM-two level | 84.73   | 92.13   |
| FRM-three level | 84.86 | 92.12   |
| FRM           | 85.16   | 92.43   |

effects on metrics, we make an ablation study and present the results in Table 5. The notation FRM-mlp, FRM-att and FRM-lm represent that the matching function mlp, att and lm are not applied to FRM, respectively. According to the results in Table 5, we can find that 1) FRM-mlp achieves the lowest accuracy, which illustrates that mlp is more important to the accuracy metric than the other two functions; 2) FRM-att achieves the lowest AUC, which illustrates that att is more important to the AUC metric than the mlp and lm functions; 3) Generally, the semantic matching functions (mlp and att) are more important to the question retrieval task than the lexical matching function (lm). This is because lexical matching suffers from the severe lexical gap problem, while semantic matching can address the lexical gap to some extent. We also notice the existence of the lexical gap problem. In addition, the performance achieved by FRM is much higher than that achieved by FRM-mlp, FRM-att and FRM-lm. The performance improvement illustrates that the combination of semantic matching and lexical matching is better than any individual, and verifies the effectiveness of the multi-view matching.

We improve the baseline BERT by incorporating the lexical matching (denoted as BM25) used in the fine-grained matching unit and the external features (denoted as ef) used in the fine-grained representation unit. Specifically, given two questions, we first use BERT to encode them and use the embedding of [CLS] to represent the encoded semantics (E_{CLS}). Second, we use BM25 model to estimate the similarity of the two questions and denote it as BM25. Third, we concatenate E_{CLS} and BM25 as the final matching representation which is the input of a classification function. For the external features, we use them to extend the semantics of a word. The results are presented in Table 5. The BERT+ef and BERT+BM25 perform better than BERT, which provides an insight that we can use external features and lexical matching to improve BERT. Besides, the performance of FRM is still better than those of ERT+ef and BERT+BM25. This illustrates that FRM exploits the external features and lexical matching more effectively than BERT+ef and BERT+BM25.

We propose the comparable pairs of keyword sets to support the fine-grained matching unit. To verify the effectiveness of multiple comparable pairs, we perform a group of experiments where different numbers of comparable pairs are applied to FRM. The results are shown in Table 6. The notation FRM-one level represents that only the highest-level keyword set pair is applied; FRM-two level (or FRM-three level) denotes that the keyword set pairs of the first two (or three) levels are applied. From Table 5, we can find that 1) FRM-three level performs better than FRM-two level and FRM-one level; 2) FRM-two level performs better than FRM-one level; 3) FRM achieves the best metrics than FRM-three level, FRM-two level and FRM-one level. In other words, the more keyword set pairs are applied, the higher the performance is achieved by FRM. These performance improvements illustrate that the comparable keyword set pairs can effectively support the multi-view matching component.

To verify the effectiveness of the fine-grained representation unit in FRM, we perform a group of experiments to investigate the effect of the deep semantic representation and external features on the performance, and present the results in Table 6. The FRM-ef denotes that the external features are not applied, and FRM-dsr denotes that the deep semantic representation is not applied. Firstly, the performance achieved by FRM-ef and FRM-dsr is lower than that achieved by FRM. This illustrates that the combination of the deep semantic representation and the external features is better than any individual for estimating the importance of keywords. Secondly, FRM-ef performs better than FRM-dsr, which indicates that the deep semantic representation is more important than external features. In addition, we use different methods to replace our proposed keyword extraction component and label the experimental results as FRM-lm and FRM-TFIDF in Table 6. The FRM-lm denotes that all keywords are assigned as equal importance, i.e., 1, and the FRM-TFIDF denotes that the TFIDF value of a keyword is used for estimating the importance. We can see that 1) the performance achieved by FRM-TFIDF is not higher than that achieved by FRM-1. This verifies that the unsupervised method can not accurately estimate the importance scores of keywords; 2) FRM performs much better than FRM-TFIDF and FRM-lm, which further verifies the effectiveness of the fine-grained representation unit.

According to the results in Table 2, 5 and 6, we find a phenomenon that some ablation models of FRM perform not better than the best baseline ERNIE, such as FRM-mlp, FRM-att, FRM-ef and FRM-dsr. The FRM-mlp and FRM-att remove the very important components from the fine-grained matching unit in FRM, i.e., MLP-based matching and attention-based matching. The FRM-ef and FRM-dsr remove the external features and the deep semantic representation feature from the fine-grained representation unit so that the multi-level keyword sets of high quality can not be constructed. So the phenomenon is in line with our expectations, and further demonstrates the importance of these components to FRM.
5 CONCLUSION

To accurately estimate the semantic similarity of two questions, we present the two aspects of the lexical gap challenge (i.e., textually distinct yet semantically equivalent and textually similar yet semantically distinct), and propose new insights of reusing important keywords to construct the fine-grained semantic representations and fine-grained matchings. To realize the insights, we use the multi-level keyword sets to model the question semantics of different granularities, propose the comparable pairs of keyword sets to support the multi-view matchings, and develop the fine-grained matching network. By experiments, we find that 1) the multi-level keyword sets can capture and model both the global and local semantics of a question, 2) the combination of semantic matching and lexical matching is more effective than any single one.

REFERENCES

[1] Li Cai, Guangyou Zhou, Kang Liu, and Jun Zhao. 2011. Learning the Latent Topics for Question Retrieval in Community QA. In Fifth International Joint Conference on Natural Language Processing, 2011. CoNLL. 1031–1037.

[2] Jing Chen, Qingcai Chen, Xin Liu, Haijun Tang, Daohu Bu, and Buzhou Tang. 2018. The BQ Corpus: A Large-scale Domain-specific Chinese Corpus For Sentence Semantic Equivalence Identification. In EMNLP. Association for Computational Linguistics, 4946–4951.

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In NAACL-HLT (1). Association for Computational Linguistics, 4171–4186.

[4] Arpita Das, Harish Yenala, Manoj Kumar Chinnakotla, and Manish Shrivastava. 2017. Fixing Weight Decay Regularization in DNNs. In ICLR.

[5] Joel R. Tetreault (Eds.). Association for Computational Linguistics, 2005. A Translation Model for Sentence Semantics of a question; 2) the combination of semantic matching and fine-grained matchings. To realize the insights, we use the multi-level keyword sets to model the question semantics of different granularities, propose the comparable pairs of keyword sets to support the multi-view matchings, and develop the fine-grained matching network. By experiments, we find that 1) the multi-level keyword sets can capture and model both the global and local semantics of a question, 2) the combination of semantic matching and lexical matching is more effective than any single one.

[6] Arpita Das, Harish Yenala, Manoj Kumar Chinnakotla, and Manish Shrivastava. 2017. Fixing Weight Decay Regularization in DNNs. In ICLR.

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In NAACL-HLT (1). Association for Computational Linguistics, 4171–4186.

[8] Arpita Das, Harish Yenala, Manoj Kumar Chinnakotla, and Manish Shrivastava. 2017. Fixing Weight Decay Regularization in DNNs. In ICLR.

[9] Joel R. Tetreault (Eds.). Association for Computational Linguistics, 2005. A Translation Model for Sentence Semantics of a question; 2) the combination of semantic matching and fine-grained matchings. To realize the insights, we use the multi-level keyword sets to model the question semantics of different granularities, propose the comparable pairs of keyword sets to support the multi-view matchings, and develop the fine-grained matching network. By experiments, we find that 1) the multi-level keyword sets can capture and model both the global and local semantics of a question, 2) the combination of semantic matching and lexical matching is more effective than any single one.

[10] Arpita Das, Harish Yenala, Manoj Kumar Chinnakotla, and Manish Shrivastava. 2017. Fixing Weight Decay Regularization in DNNs. In ICLR.

[11] Joel R. Tetreault (Eds.). Association for Computational Linguistics, 2005. A Translation Model for Sentence Semantics of a question; 2) the combination of semantic matching and fine-grained matchings. To realize the insights, we use the multi-level keyword sets to model the question semantics of different granularities, propose the comparable pairs of keyword sets to support the multi-view matchings, and develop the fine-grained matching network. By experiments, we find that 1) the multi-level keyword sets can capture and model both the global and local semantics of a question, 2) the combination of semantic matching and lexical matching is more effective than any single one.

[12] Arpita Das, Harish Yenala, Manoj Kumar Chinnakotla, and Manish Shrivastava. 2017. Fixing Weight Decay Regularization in DNNs. In ICLR.

[13] Joel R. Tetreault (Eds.). Association for Computational Linguistics, 2005. A Translation Model for Sentence Semantics of a question; 2) the combination of semantic matching and fine-grained matchings. To realize the insights, we use the multi-level keyword sets to model the question semantics of different granularities, propose the comparable pairs of keyword sets to support the multi-view matchings, and develop the fine-grained matching network. By experiments, we find that 1) the multi-level keyword sets can capture and model both the global and local semantics of a question, 2) the combination of semantic matching and lexical matching is more effective than any single one.

[14] Arpita Das, Harish Yenala, Manoj Kumar Chinnakotla, and Manish Shrivastava. 2017. Fixing Weight Decay Regularization in DNNs. In ICLR.

[15] Joel R. Tetreault (Eds.). Association for Computational Linguistics, 2005. A Translation Model for Sentence Semantics of a question; 2) the combination of semantic matching and fine-grained matchings. To realize the insights, we use the multi-level keyword sets to model the question semantics of different granularities, propose the comparable pairs of keyword sets to support the multi-view matchings, and develop the fine-grained matching network. By experiments, we find that 1) the multi-level keyword sets can capture and model both the global and local semantics of a question, 2) the combination of semantic matching and lexical matching is more effective than any single one.

[16] Arpita Das, Harish Yenala, Manoj Kumar Chinnakotla, and Manish Shrivastava. 2017. Fixing Weight Decay Regularization in DNNs. In ICLR.

[17] Joel R. Tetreault (Eds.). Association for Computational Linguistics, 2005. A Translation Model for Sentence Semantics of a question; 2) the combination of semantic matching and fine-grained matchings. To realize the insights, we use the multi-level keyword sets to model the question semantics of different granularities, propose the comparable pairs of keyword sets to support the multi-view matchings, and develop the fine-grained matching network. By experiments, we find that 1) the multi-level keyword sets can capture and model both the global and local semantics of a question, 2) the combination of semantic matching and lexical matching is more effective than any single one.

[18] Arpita Das, Harish Yenala, Manoj Kumar Chinnakotla, and Manish Shrivastava. 2017. Fixing Weight Decay Regularization in DNNs. In ICLR.

[19] Joel R. Tetreault (Eds.). Association for Computational Linguistics, 2005. A Translation Model for Sentence Semantics of a question; 2) the combination of semantic matching and fine-grained matchings. To realize the insights, we use the multi-level keyword sets to model the question semantics of different granularities, propose the comparable pairs of keyword sets to support the multi-view matchings, and develop the fine-grained matching network. By experiments, we find that 1) the multi-level keyword sets can capture and model both the global and local semantics of a question, 2) the combination of semantic matching and lexical matching is more effective than any single one.

[20] Arpita Das, Harish Yenala, Manoj Kumar Chinnakotla, and Manish Shrivastava. 2017. Fixing Weight Decay Regularization in DNNs. In ICLR.

[21] Joel R. Tetreault (Eds.). Association for Computational Linguistics, 2005. A Translation Model for Sentence Semantics of a question; 2) the combination of semantic matching and fine-grained matchings. To realize the insights, we use the multi-level keyword sets to model the question semantics of different granularities, propose the comparable pairs of keyword sets to support the multi-view matchings, and develop the fine-grained matching network. By experiments, we find that 1) the multi-level keyword sets can capture and model both the global and local semantics of a question, 2) the combination of semantic matching and lexical matching is more effective than any single one.

[22] Arpita Das, Harish Yenala, Manoj Kumar Chinnakotla, and Manish Shrivastava. 2017. Fixing Weight Decay Regularization in DNNs. In ICLR.

[23] Joel R. Tetreault (Eds.). Association for Computational Linguistics, 2005. A Translation Model for Sentence Semantics of a question; 2) the combination of semantic matching and fine-grained matchings. To realize the insights, we use the multi-level keyword sets to model the question semantics of different granularities, propose the comparable pairs of keyword sets to support the multi-view matchings, and develop the fine-grained matching network. By experiments, we find that 1) the multi-level keyword sets can capture and model both the global and local semantics of a question, 2) the combination of semantic matching and lexical matching is more effective than any single one.
[29] Kai Zhang, Wei Wu, Haocheng Wu, Zhoujun Li, and Ming Zhou. 2014. Question Retrieval with High Quality Answers in Community Question Answering. In Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management, CIKM 2014, Shanghai, China, November 3-7, 2014, Jianzhong Li, Xiaoyang Sean Wang, Minos N. Garofalakis, Ian Soboroff, Torsten Suel, and Min Wang (Eds.). ACM, 371–380. https://doi.org/10.1145/2661829.2661908

[30] Guangyou Zhou, Li Cai, Jun Zhao, and Kang Liu. 2011. Phrase-Based Translation Model for Question Retrieval in Community Question Answer Archives. In The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference, 19-24 June, 2011, Portland, Oregon, USA, Dekang Lin, Yuji Matsumoto, and Rada Mihalcea (Eds.). The Association for Computer Linguistics, 653–662. https://aclanthology.org/P11-1066/