Match^2: A Matching over Matching Model for Similar Question Identification

Zizhen Wang[*,†], Yixing Fan[†], Jiafeng Guo[*,†], Liu Yang[‡]
Ruqing Zhang[†], Yanyan Lan[*,†], Xueqi Cheng[*,†], Hui Jiang[§], Xiaozhao Wang[§]
{wangzizhen,fanyixing,guojiafeng,zhangruqing,lanyanyan,cxq}@ict.ac.cn
lyang@cs.umass.edu,{huijiang.jh,orlando}@alibaba-inc.com
[*]University of Chinese Academy of Sciences, Beijing, China
[†]CAS Key Lab of Network Data Science and Technology, Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China
[‡]University of Massachusetts Amherst, Massachusetts, United States
[§]Alibaba Group, Beijing, China

ABSTRACT

Community Question Answering (CQA) has become a primary means for people to acquire knowledge, where people are free to ask questions or submit answers. To enhance the efficiency of the service, similar question identification becomes a core task in CQA which aims to find a similar question from the archived repository whenever a new question is asked. However, it has long been a challenge to properly measure the similarity between two questions due to the inherent variation of natural language, i.e., there could be different ways to ask a same question or different questions sharing similar expressions. To alleviate this problem, it is natural to involve the existing answers for the enrichment of the archived questions. Traditional methods typically take a *one-side* usage, which leverages the answer as some expanded representation of the corresponding question. Unfortunately, this may introduce unexpected noises into the similarity computation since answers are often long and diverse, leading to inferior performance. In this work, we propose a *two-side* usage, which leverages the answer as a bridge of the two questions. The key idea is based on our observation that similar questions could be addressed by similar parts of the answer while different questions may not. In other words, we can compare the matching patterns of the two questions over the same answer to measure their similarity. In this way, we propose a novel matching over matching model, namely Match^2, which compares the matching patterns between two question-answer pairs for similar question identification. Empirical experiments on two benchmark datasets demonstrate that our model can significantly outperform previous state-of-the-art methods on the similar question identification task.

CCS CONCEPTS

• **Information systems → Question answering**.

SIGIR '20, July 25–30, 2020, Virtual Event, China

ACM ISBN 978-1-4503-8016-4/20/07…$15.00
https://doi.org/10.1145/3397271.3401143

KEYWORDS

community question answering, similar question identification; matching over matching

ACM Reference Format:
Zizhen Wang, Yixing Fan, Jiafeng Guo, Liu Yang, Ruqing Zhang, Yanyan Lan, Xueqi Cheng, Hui Jiang and Xiaozhao Wang. 2020. Match2: A Matching over Matching Model for Similar Question Identification. In *43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '20), July 25–30, 2020, Virtual Event, China.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3397271.3401143

1 INTRODUCTION

Community Question Answering (CQA) services, such as StackExchange[1] and Quora[2], have grown in popularity in recent years as a platform for people to exchange knowledge. In CQA, users can ask their questions or submit answers to questions in a collaborative fashion. Although CQA services greatly benefit users with high-quality human-generated answers for solving their problems, the efficiency becomes a big concern as the asker need to wait until someone submits the answer to his/her question. To alleviate this problem, similar question identification becomes a core task in CQA which aims to find a similar question from the archived repository whenever a new question is proposed. In the meantime, similar question identification could also help reduce redundant questions in CQA services, saving a lot of users' efforts.

However, it has long been a challenge to properly measure the similarity between two questions, which are usually very short in length, due to the inherent variation of natural language. On one hand, there could be different ways to express the same question, leading to the lexical gap [26, 33]. For example, as shown in Figure 2 Case A, the user question Q^u and the archived question Q^a are similar and could be addressed by the archived answer A^a of Q^a, but they have very different expressions. On the other hand, there could be different questions sharing very similar expressions, leading to false positive predictions if one cannot distinguish their subtle difference. For example, as shown in Figure 2 Case B, although these two questions share many words in common, their focus is totally

[1]https://stackexchange.com/
[2]https://www.quora.com/

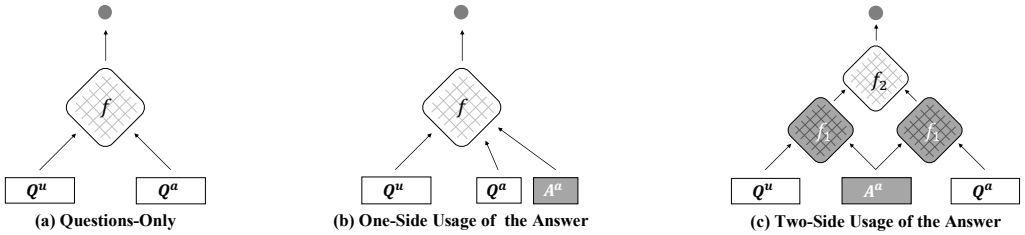

Figure 1: The architectures of similar question identification models. f denotes the identification function. The first architecture only uses the questions for identification, while the last two involve the archived answer, in which (b) treats the answer as an expand to the question and (c) leverages the answer as a bridge of the two questions.

Case A:
Q^u: What could it mean to **rotate a distribution**?
Q^a: How to understand **moments for a random variable**?
A^a: The name of moment comes from physics. Consider **the distribution of a random variable** as linear rod, the center of gravity is the first moment, and **the moment of rotational inertia about the center** of gravity is the variance.
Label: 1

Case B:
Q^u: What is the correct **method** for the Marine stutter step?
Q^a: What is the optimal Marine stutter step **timing**?
A^a : The base attack of **Marines** is **0.8608 seconds**, and **0.57387 seconds** while stimmed. The game speed factor is about 1.4. The random delay between attacks **ranges from 0.125 to -0.0625 seconds**. Based on the above numbers, we get the optimal **timing** in worst unstimmed case is **0.7041 seconds**, and **0.4992 seconds** while stimmed.
Label: 0

Case C:
Q^u: What is the **garbage collector** in Java?
Q^a: Should we avoid **object creation** in Java?
A^a: Actually, due to the memory management strategies in Java, **object creation** can be considered free compared to everything else in JVM. The other part of the cost is object destruction. The modern **garbage collector** algorithm deallocates when free memory is needed in a certain generation. If the JVM heap is big enough, then no deallocations happen long enough to cause any pauses.
Label: 0

Figure 2: The cases from StackExchange. The blue and yellow parts denote the focuses of the question and the corresponding related answer parts. The red parts denote the answer text which can address both the user question and the archived question. The archived answer is helpful to identify the question similarity (Case A,B), but it may introduce unexpected noises (Case C).

different (one about the "method" and one about the "timing"), and thus could not be addressed by the same answer.

Similar question identification has attracted extensive studies in recent years. Some early works in this direction formulated it as a question-question matching problem, as shown in Figure 1(a). Both conventional machine learning methods [7, 18, 46, 47] and deep neural networks [9, 11, 15, 31, 33, 41, 48] have been applied to this problem. However, simply based on two questions, even most advanced neural models cannot well address the two challenges mentioned above due to the sparse information in questions.

Since archived questions usually associated with answers, it is natural to involve the existing answers for the enrichment of the archived questions to alleviate the sparsity problem. To leverage the answers of the archived question, existing methods typically take a *one-side* usage, as shown in Figure 1(b), which treats the answer as some expanded representation of the corresponding question. For example, Ji et al. [26] employed the archived answer to learn an enriched topic representation of questions for similarity computation. Gupta et al. [16] matched the user question to the archived question and its answer separately then aggregated them with an attention mechanism. Unfortunately, the one-side usage may introduce unexpected noises into the similarity computation, leading to inferior performance. The reason is that answers are not equivalent representations of the corresponding questions. Answers are often long and cover diverse topics/aspects that may be beyond the scope of the corresponding question. For example, as shown in Figure 2 Case C, these two questions, one about garbage collector and one about object creation, are different in semantics. However, if we simply expand the archived question Q^a with its answer A^a which also talks about the garbage collector, we are prone to predict that these two questions are similar which is apparently a false positive prediction.

In fact, if we look at these cases carefully, we may find the following observation: similar questions could be addressed by similar parts of the answer while different questions may not. For example, as shown in Figure 2 Case A, these two questions are similar since they both can be addressed/connected by the similar parts of the archived answer A^a. However, in Case C, although the archived answer may be related to both questions, the related parts are quite different for the two questions. These cases show that it is not how similar the archived answer to the user question decides the question similarity. It is how the archived answer matches the two questions contributes to the similarity of the two questions. Therefore, we argue that the archived answer should not be simply viewed as some expansion of the corresponding question, but rather be viewed as a bridge of the two questions, namely a *two-side* usage in this work as depicted in Figure 1(c).

Based on the above idea, we propose a novel Matching over Matching Model, namely Match^2 for short, which compares the matching patterns of the two questions over the same answer for similar question identification. Specifically, Match^2 contains three modules, including the *Representation-based Similarity Module*, the *Matching Pattern-based Similarity Module* and the *Aggregation module*. The Representation-based Similarity module is similar to previous question matching methods, which generates a similarity vector between two questions simply based on their representations. The major enhancement is the Matching Pattern-based Similarity module. This module has a Siamese Network structure, which takes two question-answer pairs as the inputs, learns their matching patterns separately, builds a matching similarity tensor by comparing the

two matching patterns, and finally produces the similarity vector between the questions by compressing the matching similarity tensor. Both the representation-based and matching pattern-based similarity vectors are aggregated in the Aggregation module to produce the final identification prediction. The Aggregation module adopts a gate mechanism which takes the representation-based similarity as the primary one and the matching pattern-based similarity as the complementary one for the final decision. A multi-task learning strategy is employed to train the Match^2 model.

We evaluate the effectiveness of the proposed Match^2 model based on two widely used CQA benchmarks, i.e., CQADupStack [21] and QuoraQP[3]. To incorporate the answer information in QuoraQP, we crawled archived answers of the corresponding questions from Quora and enrich the benchmark into a new answer-expanded version, namely QuoraQP-a. The experimental results on these two benchmarks demonstrated that our method can significantly outperform those state-of-the-art methods on the similar question identification task.

The major contributions of this paper include:

(1) We analyze the role of the archived answer in the similar question identification task and propose a two-side usage of the answer which leverages it as a bridge of the two questions.
(2) We propose a novel matching over matching (Match^2) model to compare the matching patterns of the two questions over the same answer for similar question identification.
(3) We conduct extensive comparisons and analysis against the state-of-the-art similar question identification models on benchmarks to demonstrate the effectiveness of our proposed method.

2 RELATED WORK

In this section, we briefly review the most related topics to our work in CQA, i.e., question matching. Question matching which evaluates the similarity between two questions, could be further divided into the question deduplication task and the similar question identification task with regard to different application scenarios.

2.1 Question Deduplication

Question deduplication aims to merge or remove the redundant questions in the archived question threads. Early studies mainly focused on designing effective features to measure the similarities between two questions, such as lexical features [4, 17, 23], syntactic features [8, 30, 42], or heuristic features [3, 13]. Many recent successes on this task have been achieved by advanced neural network models. For example, Pang et al. [32] evaluated the question similarity from hierarchical levels. Wan et al. [41] modeled the recursive structure between question pairs with spatial RNN. Tay et al. [38] proposed a CSRAN model to learn fine-grained question matching details. Yang et al. [48] built RE2 model with stacked alignment layers to keep the model fast while still yielding strong performance, and Devlin et al. [11] pre-trained a stacked transformer network which can be used for question deduplication task after fine-tuning.

Besides, the question threads in the community include not only the question texts but also other information, e.g., topics, comments and answers, which provide other perspectives for question deduplication. Zhang et al. [51] proposed a topic model approach to take answer quality into account. Wu et al. [45] proposed the QCN network to make use of the subject-body relationship of the community questions. Filice et al. [13] proposed a method to utilize the interconnection information between the question and its comments. Liang et al. [27] employed adaptive multi-attention mechanism to enhance questions with their corresponding answers. Moreover, many researchers have considered the use of different kinds of external resources. Wu et al. [44] employed various types of handcraft features to measure the question semantic similarity. Zhou et al. [54] used the semantic relations extracted from the global knowledge of WikiPedia[4].

2.2 Similar Question Identification

Similar question identification aims to find a similar question from the archived repository for a new question issued by a user. It is usual to frame the similar question identification as a retrieval task where the user question is taken as a query and archived questions are ranked based on their semantic similarities to the query. Hence, classical retrieval methods, e.g., BM25 [34] and LMIR [50], have beed applied for this task. There are also researchers employed statistical translation [25, 46, 52], topic model [5, 55] and relation extraction methods [35] to identify the similar questions. Recently, deep learning methods have been widely adopted to solve it. For example, Qiu et al. [33] employed convolutional neural network to encode questions in semantic space. Wan et al. [40] proposed MV-LSTM to capture the contextualized local information with multiple positional question representations. Furthermore, many works considered the use of different kinds of complementary information, such as question category [6, 12, 53], Wikipedia concepts [2] and corresponding answer [16, 26, 36].

Even some of the researchers on similar question identification have focused on ranking models, they might face the computational complexity and evaluation difficulty problem [20]. To address this issue, many works model the task as a classification task, which aims to explicitly predict whether the archived question is similar with the user question or not. For example, Wang et al. [43] employed a bilateral mechanism to enhance single direction matching. Chen et al. [9] proposed a sequential inference model based on chain LSTMs for the recursive matching architectures. Gong et al. [15] used DenseNet [22] to hierarchically extract semantic features from questions interaction space. Hoogeveen et al. [20] adopted meta data such as user features to identify the question relation. It seems that some models could not only be applied to similar question identification but also question deduplication task, but we can find the clear difference, i.e., the user question in similar question identification has few information except the text itself.

3 OUR APPROACH

In this section, we present the Matching over Matching (Match^2) model for the similar question identification task in detail. We first

[3]https://www.kaggle.com/c/quora-question-pairs

[4]https://en.wikipedia.org/wiki/Main_Page

Table 1: A summary of key notations in this work.

$\mathbf{P}^u, \mathbf{P}^a$	The matching pattern of the user question and the archived question over the archived answer
$\mathbf{P}^s$	The pattern similarity tensor
$\mathcal{S}, S, s$	The pattern similarity function at tensor-, layer- and element-wise
$\mathbf{v}_q$	The representation-based similarity vector
$\mathbf{v}_a$	The matching pattern-based similarity vector
$\mathbf{v}$	The question similarity vector
r	The main task loss ratio

give an overview of the problem formulation and model architecture, and then describe each module of our model as well as the learning procedure. A summary of key notations in this work is presented in Table 1.

3.1 Overview

Formally, given a user question Q^u, an archived question Q^a, and an answer A^a of the archived question, Match^2 aims to learn a classification model $f(\cdot)$ to predict the similarity score y^q between the user question Q^u and the archived question Q^a.

Basically, our Match^2 model contains the following components: (1) Representation-based Similarity module: to produce a similarity vector between the two questions based on their representations; (2) Matching Pattern-based Similarity module: to compare the matching patterns of the two questions over the archived answer; and (3) Aggregation module: to produce the final similarity score by aggregating the representation-based and matching pattern-based module. The overall architecture is depicted in Figure 3 and we will detail our model as follows.

3.2 Representation-based Similarity Module

Generally, the representation-based similarity module takes the two questions as inputs and predicts a similarity vector, which is similar to previous question matching methods. In this work, we adopt the Bert [11] to measure the question similarity due to its superiority in many natural language understanding tasks.

Firstly, we concatenate the questions to the required format, which starts with a $[CLS]$ token for the whole sequence representation and ends with a $[SEP]$ token to denote the separator boundary of each question. Then, we use the stacked transformer architecture to encode the formatted questions to obtain the representations. Specifically, after being embedded, the input is processed by a multi-head attention network and a feed forward network in each transformer layer. This stacked structure has two types of outputs,

$$\mathbf{B}^p, \mathbf{B}^s = StackedTransformer(Q^u, Q^a), \tag{1}$$

where $\mathbf{B}^p$ denotes the pooled feature corresponded to $[CLS]$ and $\mathbf{B}^s$ represents the sequence features of the whole input sequence. We adopt $\mathbf{B}^p$ as the representation-based similarity vector $\mathbf{v}_q = \mathbf{B}^p \in R^H$, where H is the hidden size of Bert.

3.3 Matching Pattern-based Similarity Module

The matching pattern-based similarity module is responsible for contrasting the matching patterns of the two questions over the same answer. As shown in the right part of Figure 3, this module has a Siamese Network structure, which includes three dependent layers: (1) matching pattern layer: to take two question-answer pairs as the inputs and learns their matching patterns separately; (2) pattern similarity layer: to build a pattern similarity tensor by contrasting the two matching patterns; and (3) compression layer: to produce the similarity vector between the questions by compressing the matching similarity tensor.

3.3.1 ***Matching Pattern Layer.*** We adopt Bert again to compute the matching patterns of the two questions over the same answer. Different from the $\mathbf{B}^p$ from Equ.1, we use the sequence features $\mathbf{B}^s$ here and divide it into two parts to represent the question and answer respectively.

Take the matching pattern $\mathbf{P}^u$ between the user question Q^u and the archived answer A^a as an example. Firstly, the user question Q^u with a sequence of m tokens is represented by concatenating the question sequence features from each transformer layer,

$$\widehat{\mathbf{Q}}^u = [\widehat{\mathbf{Q}}^u_1, \cdots, \widehat{\mathbf{Q}}^u_l, \cdots, \widehat{\mathbf{Q}}^u_L],$$

where L is the number of transformer layers in Bert, $\widehat{\mathbf{Q}}^u_l \in R^{H\times m}$ is the l-th question sequence feature separated from $\mathbf{B}^s$. In the same way, the archived answer A^a that has w tokens is represented as $\widehat{\mathbf{A}}^a \in R^{L\times H\times w}$. Finally, the layer-wise matching pattern $\mathbf{P}^u_l$ between Q^u and A^a in the l-th transformer layer is computed as,

$$\mathbf{P}^u_l = \widehat{\mathbf{Q}}^{uT}_l \widehat{\mathbf{A}}^a_l.$$

Hence, by concatenating the L layer-wise matching patterns, we can obtain the final matching pattern $\mathbf{P}^u$ between the user question Q^u and the archived answer A^a, i.e.,

$$\mathbf{P}^u = [\mathbf{P}^u_1, \mathbf{P}^u_2, \cdots, \mathbf{P}^u_L] \in R^{L\times m\times w}$$

.

The matching pattern $\mathbf{P}^a$ of the archived question answer pair can be computed in the same way as described above. It should be noted that the Bert architecture in this module does not share the parameters with that used in Section 3.2.

3.3.2 ***Pattern Similarity Layer.*** In this layer, we compute a pattern matching similarity tensor $\mathbf{P}^s$ given the two matching patterns $\mathbf{P}^u$ and $\mathbf{P}^a$, i.e.,

$$\mathbf{P}^s = \mathcal{S}(\mathbf{P}^u, \mathbf{P}^a) \in R^{L\times m\times n},$$

where $\mathcal{S}$ denotes the tensor-wise similarity function, and $\mathbf{P}^s_l$ denotes the layer-wise matching pattern which is defined as,

$$\mathbf{P}^s_l = S(\mathbf{P}^u_l, \mathbf{P}^a_l) \in R^{m\times n}.$$

Specifically, the element-wise matching pattern similarity scalar $P^s_{l,ij}$ is computed by,

$$P^s_{l,ij} = s(\mathbf{P}^u_{l,i}, \mathbf{P}^a_{l,j}),$$

where $\mathbf{P}^u_{l,i}$ is the matching pattern from the i-th token in the user question to the archived answer, as well as $\mathbf{P}^a_{l,j}$ represents that from the j-th archived question token.

Here, we propose five element-wise similarity functions $s(\mathbf{x}, \mathbf{y})$ to compute the similarity between a question and an answer.

- *Dot product* between two vectors is based on the projection of one vector onto another, which is defined as follows:

$$s_{dot}(\mathbf{x}, \mathbf{y}) =< \mathbf{x}, \mathbf{y} >= \mathbf{x}^T \mathbf{y}.$$

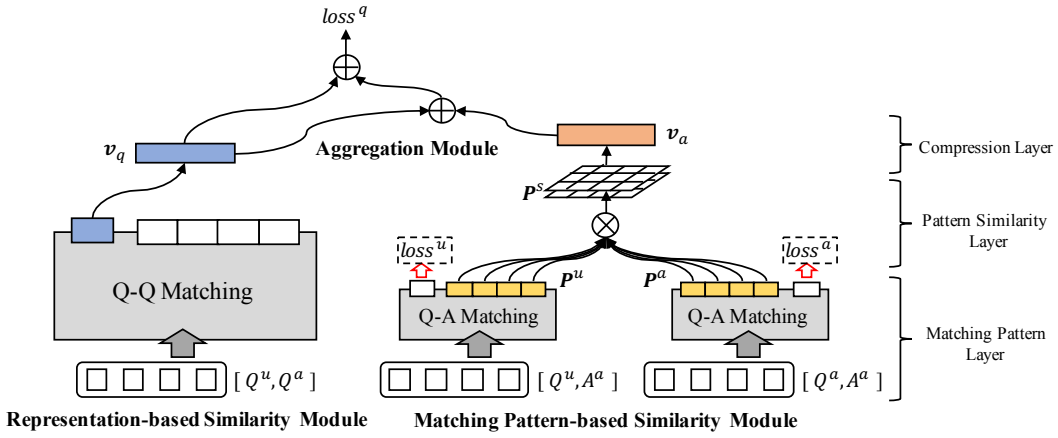

Figure 3: The architecture of the Matching over Matching Model.

- *Cosine* is a common function to model interactions. The similarity score is viewed as the angle of two vectors:

$$s_{cos}(\mathbf{x}, \mathbf{y}) = \frac{< \mathbf{x}, \mathbf{y} >}{||\mathbf{x}||||\mathbf{y}||}$$

where $||\cdot||$ denote the L_2 norm of vector.

- *L1* [24] represents the similarity based on Manhattan distance between vectors as follows,

$$s_{l1}(\mathbf{x}, \mathbf{y}) = \frac{1}{1 + \Sigma_{t=1}|\mathbf{x}_t - \mathbf{y}_t|}.$$

- *L2* is another widely used distance-based similarity function. Different from the *L1* function, it is based on euclidean distance, namely,

$$s_{l2}(\mathbf{x}, \mathbf{y}) = \frac{1}{1 + \sqrt{\Sigma_{t=1}(\mathbf{x}_t - \mathbf{y}_t)^2}}.$$

- *Jesene-Shannon* [24] firstly transforms the vector to a distribution with $softmax$ function, and then quantifies their difference by Jesene-Shannon Divergence [14],

$$s_{jss}(\mathbf{x}, \mathbf{y}) = 1 - JSD(softmax(\mathbf{x}), softmax(\mathbf{y})).$$

3.3.3 ***Compression Layer.*** The compression layer aims to produce the matching pattern-based similarity vector by compressing the pattern similarity tensor $\mathbf{P}^s$ to a low dimension vector. We firstly use a two-layer BN-ReLU-Conv [19] structure with H filters to introduce contextual information, and then adopt the average global pooling method [28] to obtain the final matching pattern-based similarity vector $\mathbf{v}_a \in R^H$.

3.4 Aggregation Module

The similarity vectors from previous two modules are combined to compute the question similarity score y^q in this module. Given $\mathbf{v}_q$ and $\mathbf{v}_a$, we introduce a gate mechanism inspired by GRU [10], which takes the former as the primary one and the latter as the complementary one, to obtain the final question similarity vector $\mathbf{v}$. Specifically, it can be computed by

$$\begin{aligned}
\mathbf{r} &= \sigma(\mathbf{W}_r\mathbf{v}_a + \mathbf{U}_r\mathbf{v}_q), \\
\mathbf{z} &= \sigma(\mathbf{W}_z\mathbf{v}_a + \mathbf{U}_z\mathbf{v}_q), \\
\widehat{\mathbf{v}} &= tanh(\mathbf{W}\mathbf{v}_a + \mathbf{U}(\mathbf{r} \otimes \mathbf{v}_q)), \\
\mathbf{v} &= \mathbf{z}\mathbf{v}_q + (1 - \mathbf{z})\widehat{\mathbf{v}},
\end{aligned}$$

where $\otimes$ is the element-wise multiplication, σ denotes the sigmoid function, and $\mathbf{W}_r, \mathbf{W}_z, \mathbf{W}, \mathbf{U}_r, \mathbf{U}_z, \mathbf{U}$ are trainable parameters.

Based on the question similarity vector $\mathbf{v}$, we then apply a multi-layer perceptron (MLP) to obtain the question similarity score y^q,

$$y^q = \sigma(\mathbf{W}_2 ReLU(\mathbf{W}_1\mathbf{v} + \mathbf{b}_1) + \mathbf{b}_2), \tag{2}$$

in which $\mathbf{W}_1$, $\mathbf{b}_1$, $\mathbf{W}_2$ and $\mathbf{b}_2$ are trainable parameters.

3.5 Model Training and Inference

In the training phase, we employ the cross-entropy loss to learn our Match^2 model in an end-to-end way. To train the model sufficiently, we adopt a multi-task learning strategy to combine the question-question matching task and the question-answer matching task. The question-question matching task aims to measure the similarity between two questions as our main task, while the question-answer matching is an auxiliary task that aims to evaluate whether the answer can satisfy the question in the matching pattern-based similarity module. For the auxiliary task, we employ the $\mathbf{B}^p$ (see Equ. 1) from the matching pattern layer for prediction. We apply a multi-layer perceptron described in Equ. 2 to calculate the similarity score y^u between the user question and the archived answer. In the same way, we get y^a to represent the archived question answer pair similarity score. However, due to the lack of question-answer matching labels, we should build the ground-truth for the auxiliary task. In details, (1) for each archived question, we regard the corresponding archived answer as the relevant answer; (2) for each user question, we regard the corresponding answer with respect to its similar question as the relevant answer. Thus, we computed the question-answer matching loss $loss^u$ and $loss^a$ with cross-entropy loss again. The overall loss is defined as the weighted sums of three losses, i.e.,

$$loss = r loss^q + \frac{1-r}{2} loss^u + \frac{1-r}{2} loss^a,$$

Table 2: Dataset statistics. # denotes the number of instances, $|len_Q|$ and $|len_A|$ denote the average length of the questions and answers, respectively.

	#Train	#Dev	#Test	$\|len_Q\|$	$\|len_A\|$
CQADupStack	56,633	5000	5000	11.89	177.70
QuoraQP-a	281,480	10,000	10,000	13.83	45.65

where $r \in [0, 1]$ is the main task loss ratio. To overcome the issue of sparse irrelevant answers, for each question, we random sample irrelevant answers from its top-K[5] candidate answers which are retrieved from the whole answer collection by BM25 [34] method. Note if the answer is irrelevant to both the user question and the archived question, we set r as 0 while training this instance because the answer can not be a bridge in this situation.

In the inference phase, given the user question Q^u, the archived question Q^a and the real archived answer A^a, we compare the prediction y^q with the threshold 0.5 to identify whether the questions are similar or not.

4 EXPERIMENTS

4.1 Datasets

We evaluate our model on the following two datasets, i.e., CQADupStack and QuoraQP-a (answer-expanded version of QuoraQP). The datiled statistics of these datasets are shown in Table 2.

- **CQADupStack** is a benchmark dataset which is widely used in CQA [21]. It contains question threads sampled from twelve StackExchange subforums and annotated with similar question information. We take the annotated best answer of the question as the archived answer. If there is no best answer for the question, we directly use the answer with the highest score as the archived answer.
- **QuoraQP-a** is built on the widely used CQA dataset QuoraQP[6], which contains 537,933 distinct questions from Quora. The original dataset cannot be used for our task directly since it does not include archived answers. To evaluate our model, we randomly select one question in each pair as the user question and set another one as the archived question. Then, we take the top ranked answer from the original website[7] as the archived answer[8].

4.2 Baseline Models

We compare our proposed model with previous similar question identification methods, which could be classified into two categories based on the usage of answers, i.e., question-only methods and one-side methods.

4.2.1 ***Question-only Methods.*** Here we consider six existing methods which only rely on questions for similar question identification.

- **TSUBAKI** [37] accounts for a dependency structure of a sentence and synonyms to evaluate the question similarity.
- **BiMPM** [43] employs a bilateral mechanism to enhance single direction matching in sentence pair relevance modeling.
- **ESIM** [9] is a sequential inference model based on chain LSTMs, which considers the recursive architectures in both local inference modeling and inference composition.
- **DIIN** [15] is a instance of Interactive Inference Network (IIN) architecture that hierarchically extracting semantic features from interaction space.
- **RE2** [48] is a fast and strong neural model with stacked alignment layers, which also employ fusion layer to make the model deeper.
- **Bert** [11] is a pre-trained language model based on stacked Transformer [39] layers, which is effective in measuring the text pair similarity.

4.2.2 ***One-side Methods.*** We also consider recently proposed methods that employ one-side usage of the archived answer for similar question identification.

- **TSUBAKI+Bert** [36] is a recently proposed method that combine the similarity between questions and the relevance between the user question and archived answer.

Here, to fully demonstrate the effectiveness of our model, we also incorporate answers into those question-only methods by two basic operators [16]. The first one is to directly concatenate the archived question along with its answer. we denote these methods as M_{concat}, where M could be any method in the question-only Methods. The second one is M_{attn}, which effectively combines the similarity representations from both the question pair matching and the question answer matching using attention mechanisms in a hierarchical manner.

4.3 Implementation Details

We implement our model by Tensorflow [1]. The hyper-parameters are tuned with the development set. The model is trained end-to-end by RAdam [29] optimizer. We set the learning rate of RAdam as $5e-5$, and other parameters as $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 1e-6$. We use 3×3 kernels in the compression layer. We use an exponential decayed keep rate during training, where the initial keep rate is 1.0 and the decay rate is 0.933 for every 5000 steps, where the keep rate will achieve to 0.5 after 50,000 steps. We initial the Bert structure in our model with released Bert-base model[9]. The other parameters are randomly initialized under a normal distribution with $\mu = 0$ and $\sigma = 0.2$. The maximum question length is truncated to 24 for CQADupStack and 32 for QuoraQP-a. The maximum answer length is truncated to 256 for CQADupStack and 100 for QuoraQP-a. The batch size is 32 for CQADupStack and 48 for QuoraQP-a.

For evaluation, we adopt Accuracy, Precision, Recall and F1 score to evaluate the models, and set the Accuracy as the main metric.

4.4 Hyper-parameter Analysis

4.4.1 ***Pattern Similarity Function.*** As described in Section 3.3.2, we can adopt various pattern similarity functions to calculate the pattern similarity tensor. Here, we study the performance of five candidate functions. The results are shown in Table 3. As we can see, the choice of pattern similarity function does affect the performance of the Match^2 model. Specifically, the *dot* function has

[5]We set $K = 5$ in this paper.
[6]https://www.kaggle.com/c/quora-question-pairs
[7]https://www.quora.com/
[8]We released the dataset at http://tinyurl.com/y8kbbfyu
[9]https://storage.googleapis.com/bert_models/2018_10_18/uncased_L-12_H-768_A-12.zip

Table 3: Results of different similarity functions in the matching pattern-based module on CQADupStack.

Function	Accuracy	Precision	Recall	F1
dot	**62.84**	**56.34**	**55.12**	**55.72**
cos	62.80	56.29	55.07	55.67
l1	62.44	55.90	54.31	55.09
l2	62.46	55.89	54.55	55.21
jss	62.76	56.25	54.97	55.60

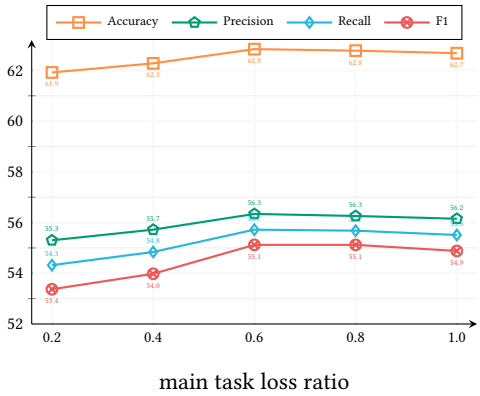

Figure 4: Results of different main task loss ratios of Match^2 on CQADupStack.

achieved the best performance in terms of all the evaluation metrics. The reason might be that the *dot* function could capture the detailed interactions of each dimension, which are useful in identifying the similarities between matching patterns. In the following experiments, we will use the *dot* as the pattern similarity function.

4.4.2 ***Multi-task analysis.*** In the model learning phase, we introduced an additional task to train the model. The final optimization objective of the model is the linear combined loss with pre-defined main task loss ratio r. Here, we study how this ratio affects the model performance. Specifically, we set the weight value from 0.2 to 1.0, where the larger value denotes more emphasis on the main task, i.e., similar question identification task. The results are depicted in Figure 4, we can see that there is a consistent tendency between all the evaluation metrics, i.e., the performance first improves along with the increase of the weight value, and drops when the weight become larger than 0.6. The best performance can be obtained at 0.6, where the model pays balanced attention to both learning objectives.

4.5 Main Results

In this section, we show the main results of the Match^2 model as well as baseline methods. All the results are summarized in Table 4.

Firstly, for the question-only methods, we can see that neural models (e.g., BiMPM, ESIM and etc.) achieve significant better performance than traditional methods (i.e., TSUBAKI) on both datasets. Moreover, it can be observed that the relative improvement of the neural methods over TSUBAKI is much larger on QuoraQP-a than the CQADupStack. The reason might that the QuoraQP-a is much larger in size than the CQADupStack, where neural models are often data hungry. The Bert achieves the best performance on both datasets in terms of all metrics. This indicates the models pre-trained on a large amounts of unstructured texts learn to encode linguistic features that improve the performance.

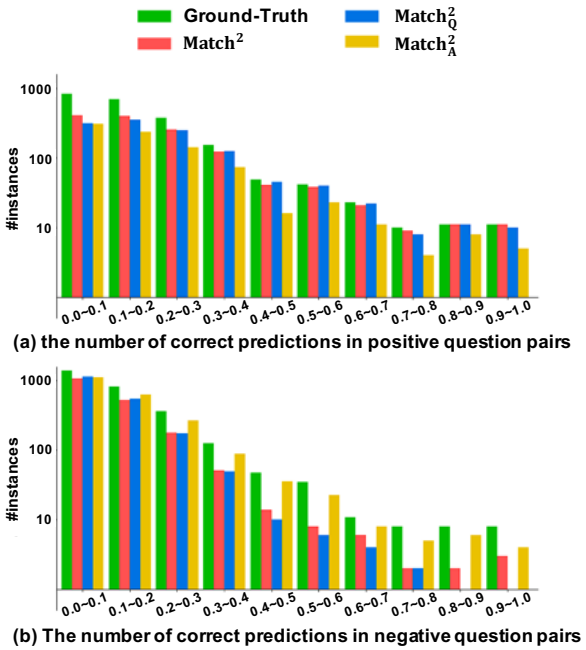

Figure 5: Results on different question groups of CQADupStack. The x-axis represents the Jaccard Index.

Secondly, comparing the one-side methods with the question-only methods, we can find that incorporating the answers could indeed improve the performance. However, there are also some methods achieving inferior performance with the archived answer. For example, the accuracy of RE$_{concat}$ decreases from 60.56 to 60.16 on the CQADupStack. This demonstrates that simply incorporating the answers could introduce unexpected noises, which could possibly hurt the performance. Moreover, we find that the attention method is relatively more effective than the concatenation method, which indicates the possibility to improve the performance by carefully designed answer usage method.

Thirdly, the Match^2 model achieves the best performance in terms of all metrics on both benchmarks. For example, the relative improvement of the Match^2 model over the best performing baseline method (i.e., Bert$_{attn}$) is about 3.3% and 1.3% in terms of F1 metric on CQADupStack and QuoraQP-a. All these demonstrate the effectiveness of the Match^2.

4.6 Analysis on the Match^2

To better analyze the effect of different components in Match^2, we first construct three variants of the model, then evaluate them on both benchmarks and on different question groups. The constructed variants are listed as follows:

- **Match^2_Q** is used to represent the representation-based module. It removes the matching pattern-based module and use a multi-layer perceptron (MLP) to replace the aggregation module.

Table 4: Main Results on CQADupStack and QuoraQP-a. †indicates the statistically significant difference over the best baseline model, where +/- indicates the statistically significant improvement/deterioration over the question-only counterpart with $p < 0.01$ [49] .

			CQADupStack				QuoraQP-a			
Method	#	Model	Accuracy	Precision	Recall	F1	Accuracy	Precision	Recall	F1
question-only	1	TSUBAKI	56.20	50.17	34.12	40.62	66.78	51.61	36.25	42.59
	2	BiMPM	59.44	54.84	43.1	48.27	87.28	81.59	80.82	81.20
	3	ESIM	58.64	53.85	40.46	46.20	86.86	80.78	80.49	80.64
	4	DIIN	60.30	56.13	43.83	49.22	88.01	82.48	82.20	82.33
	5	RE2	60.56	56.47	44.33	49.67	88.30	82.85	82.70	82.77
	6	Bert	60.92	56.91	45.24	50.41	89.24	84.21	84.11	84.16
one-side	7	TSUBAKI+Bert	57.20^{+}	51.75^{+}	37.13^{+}	43.24^{+}	80.23^{+}	70.89^{+}	70.99^{+}	70.94^{+}
	8	BiMPM_{concat}	59.48	54.55	46.15^{+}	50.00^{+}	86.70^{-}	80.15^{-}	80.91^{+}	80.53^{-}
	9	ESIM_{concat}	59.05^{+}	54.35	42.14^{+}	47.47^{+}	86.66^{-}	80.09^{-}	80.85^{+}	80.47^{-}
	10	DIIN_{concat}	60.74^{+}	56.31^{+}	47.15^{+}	51.33^{+}	88.25^{+}	82.63	82.85^{+}	82.74^{+}
	11	RE2_{concat}	60.16^{-}	54.96^{-}	51.21^{+}	53.02^{+}	87.71^{-}	79.53^{-}	85.91^{+}	82.62^{-}
	12	Bert_{concat}	61.50^{+}	56.70	52.27^{+}	54.26^{+}	89.81^{+}	84.35	85.97^{+}	85.15^{+}
	13	BiMPM_{attn}	59.74	55.11	44.69^{+}	49.36^{+}	88.18^{+}	83.28^{+}	81.61^{+}	82.44^{+}
	14	ESIM_{attn}	59.38^{+}	54.93^{+}	41.59^{+}	47.34^{+}	87.82^{+}	82.76^{+}	81.05^{+}	81.90^{+}
	15	DIIN_{attn}	60.78^{+}	56.84^{+}	44.28	49.78	88.72^{+}	82.76^{+}	84.08^{+}	83.52^{+}
	16	RE2_{attn}	61.18^{+}	56.55	49.93^{+}	53.04^{+}	89.07^{+}	83.32^{+}	84.82^{+}	84.06^{+}
	17	Bert_{attn}	61.96^{+}	57.31^{+}	52.35^{+}	54.71^{+}	89.92^{+}	85.13^{+}	85.23^{+}	85.18^{+}
two-side	18	Match^2	$\mathbf{62.78}^{\dagger}$	$\mathbf{58.02}^{\dagger}$	$\mathbf{55.03}^{\dagger}$	$\mathbf{56.49}^{\dagger}$	$\mathbf{90.65}^{\dagger}$	$\mathbf{86.21}^{\dagger}$	$\mathbf{86.29}^{\dagger}$	$\mathbf{86.25}^{\dagger}$

Table 5: Ablation results on CQADupStack and QuoraQP-a. †indicates the statistically significant difference over the Match^2 model with $p < 0.01$ [49].

	CQADupStack				QuoraQP-a			
Model	Accuracy	Precision	Recall	F1	Accuracy	Precision	Recall	F1
Match^2	62.78	58.02	55.03	56.49	90.65	86.21	86.29	86.25
Match^2_Q	$60.94^{\dagger}$	$56.92^{\dagger}$	$45.33^{\dagger}$	$50.47^{\dagger}$	$89.24^{\dagger}$	$84.21^{\dagger}$	$84.11^{\dagger}$	$84.16^{\dagger}$
Match^2_A	$60.44^{\dagger}$	$55.46^{\dagger}$	$50.25^{\dagger}$	$52.72^{\dagger}$	$89.11^{\dagger}$	$83.32^{\dagger}$	$84.97^{\dagger}$	$84.14^{\dagger}$
Match^2_{attn}	$62.32^{\dagger}$	58.00	$51.34^{\dagger}$	$54.47^{\dagger}$	$90.04^{\dagger}$	$85.42^{\dagger}$	$85.26^{\dagger}$	$85.34^{\dagger}$

- $\textbf{Match}^2_A$ is for matching pattern-based module. It removes the representation-based module and use a MLP for aggregation.
- $\textbf{Match}^2_{attn}$ adopts the attention mechanism [16] to replace the gate mechanism in the aggregation module,to analyze the effect of the gate mechanism.

4.6.1 ***Sub-Module Analysis.*** The performance of different variants are shown in Table 5. Firstly, we can see that both of the Match^2_Q and Match^2_A achieve relatively good performance with the sub-module itself, which demonstrates these modules are effective in most cases. Secondly, comparing with these two variants, we find that Match^2_Q achieves higher precision while Match^2_A achieves higher recall. This indicates the representation-based module and the matching-pattern based module could be complementary to each other. Finally, we observe the attention mechanism cannot fully utilize the advantages of the previous modules, which is particularly reflected in the recall metric. This difference demonstrates the effectiveness of the gating mechanism in the aggregation component.

4.6.2 ***Question Group-based Analysis.*** For more detailed analysis of model performance, we divide the question pairs in CQADupStack into twenty groups based on their similarity and Jaccard Index [23], which is a widely-used word level similarity feature. We analyze the number of correct predictions in each group. The results are shown in Figure 5.

We notice that the positive and negative question pairs have similar Jaccard Index distribution in CQADupStack. Specifically, for the positive question pairs, we can see the Match^2_Q achieves better performance than the Match^2_A on all the groups, and the gap is larger on the pairs with higher Jaccard Index, i.e. more shared words. It indicates that the Match^2_Q could directly capture the word similarities, which is useful to the similar questions with many shared words.

Table 6: Two cases from the CQADupStack data. Match^2_Q is the representation-based similarity module, and Match^2_A is the matching pattern-based similarity module.

		Ground-truth	Match^2_Q	Match^2_A	Match^2
Case I	Q^u: how to keep a session when logging out Q^a: can i close a terminal without killing the command running in it A^a: once you log out a terminal, this kill the running session in it as well. to keep the session alive, you should start a session with 'nohup' command. another way is pause the session with 'ctrl-z', pull it into the background with 'bg' and then 'disown' it.	1	0	1	1
Case II	Q^u: how do i clean up radiation Q^a: how long does radiation take to clean up A^a: you have to wait more than 20 years for all the radiation to turn into ground pollution	0	1	0	0

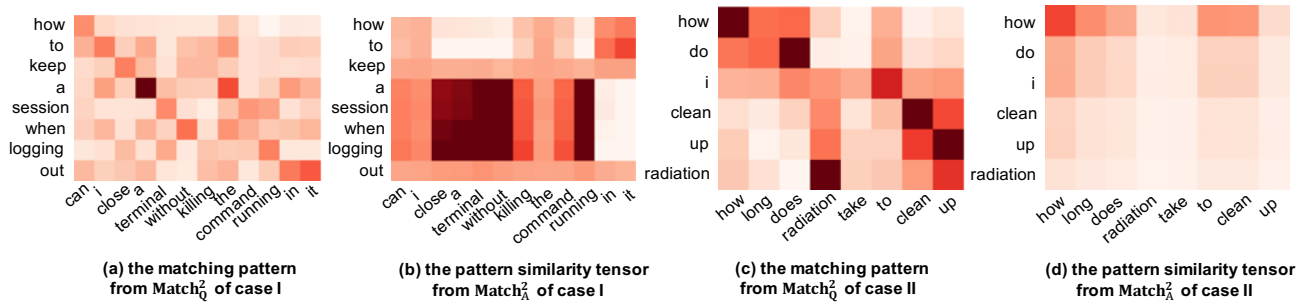

Figure 6: Visualization of the matching pattern and pattern similarity tensor of the above two cases, deeper color indicates higher similarity.

On the other hand, for the negative question pairs, we can observe that the Match^2_Q could not well address the negative questions pairs with higher Jaccard Index. For example, when the Jaccard Index higher than 0.8, the Match^2_Q fails on all the instances. The Match^2_A outperforms the Match^2_Q especially on the higher Jaccard Index groups. This demonstrates that the matching pattern could avoid the noises from shared words and emphasize the difference between questions. Finally, the Match^2 module could outperform these two types of modules in most cases. It indicates the effectiveness of the gate mechanism that combines the advantage of these two module into a unified model.

4.7 Case Study and Visualization

Here, we conduct case studies to better understand what have been learned by the Match^2 model. We also take the Match^2_Q and Match^2_A for comparison. The instances are shown in Table 6, the first one is a positive question pair with few shared words, while the second one is a negative pair with more common words. We can see that the Match^2_Q is not good at dealing with these types of questions, but the Match^2 could correctly identify them with the help of Match^2_A. Specifically, we visualize the matching patterns and pattern similarity tensors from Match^2_Q and Match^2_A in Figure 6. For case I, we notice the Match^2_Q is difficult to find out the semantic relation between questions but only recognizes the cluttered similarity presented in Figure 6(a). By leveraging the archived answer as a bridge, the Match^2_A can easily identify the similarities by comparing the matching patterns, as shown in Figure 6(b).

For case II, we notice that the Match^2_Q highlights three similar phrases in Figure 6(c), and makes a false positive prediction that the questions are similar. On the other hand, as shown in Figure 6(d), the only semantic relevance between two questions is "how" which means the questions are different except their question type. Based on the pattern similarity tensor, the Match^2_A is able to predict these two questions as different.

5 CONCLUSION AND FUTURE WORK

In this paper, we introduced a two-side usage of the archived answer for similar question identification task by leveraging the answer as a bridge of the questions. We proposed a novel matching over matching (Match^2) model, which consists of three main components, namely the *representation-based similarity module*, *matching pattern-based similarity module*, and the *aggregation module*. Empirical experiments on two benchmarks demonstrate that our model can significantly outperform previous state-of-the-art methods. Moreover, we also conducted rigorous experiments on the sub-modules to verify the effectiveness of the model. In the future work, we would like to extend our model to leverage variant number of answers and take the answer quality into account.

ACKNOWLEDGMENTS

This work was supported by the National Natural Science Foundation of China (NSFC) under Grants No. 61722211, 61773362, 61872338, and 61902381, Beijing Academy of Artificial Intelligence (BAAI) under Grants No. BAAI2019ZD0306, and BAAI2020ZJ0303, the Youth Innovation Promotion Association CAS under Grants No. 20144310, and 2016102, the National Key RD Program of China under Grants No. 2016QY02D0405, the Lenovo-CAS Joint Lab Youth Scientist Project, and the Foundation and Frontier Research Key Program of Chongqing Science and Technology Commission (No. cstc2017jcyjBX0059).

REFERENCES

[1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, et al. 2016. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv* (2016).

[2] Muhammad Ahasanuzzaman, Muhammad Asaduzzaman, Chanchal K Roy, and Kevin A Schneider. 2016. Mining duplicate questions in stack overflow. In *MSR*. ACM, 402–412.

[3] Alberto Barrón-Cedeno, Simone Filice, Giovanni Da San Martino, Shafiq Joty, Lluís Màrquez, Preslav Nakov, and Alessandro Moschitti. 2015. Thread-level information for comment classification in community question answering. In *ACL*. 687–693.

[4] Andrei Z Broder. 1997. On the resemblance and containment of documents. In *Proceedings. Compression and Complexity of SEQUENCES*. IEEE, 21–29.

[5] Li Cai, Guangyou Zhou, Kang Liu, and Jun Zhao. 2011. Learning the Latent Topics for Question Retrieval in Community QA. In *IJCNLP*.

[6] Xin Cao, Gao Cong, Bin Cui, Christian S Jensen, and Quan Yuan. 2012. Approaches to exploring category information for question retrieval in community question-answer archives. *TOIS* 30, 2 (2012), 7.

[7] Yunbo Cao, Huizhong Duan, Chin-Yew Lin, Yong Yu, and Hsiao-Wuen Hon. 2008. Recommending questions using the mdl-based tree cut model. In *WWW*. ACM, 81–90.

[8] David Carmel, Avihai Mejer, Yuval Pinter, and Idan Szpektor. 2014. Improving term weighting for community question answering search using syntactic analysis. In *CIKM*. ACM, 351–360.

[9] Qian Chen, Xiaodan Zhu, Zhenhua Ling, Si Wei, Hui Jiang, and Diana Inkpen. 2016. Enhanced lstm for natural language inference. *arXiv* (2016).

[10] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv* (2014).

[11] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *NAACL* (2018).

[12] Huizhong Duan, Yunbo Cao, Chin-Yew Lin, and Yong Yu. 2008. Searching questions by identifying question topic and question focus. In *ACL*. 156–164.

[13] Simone Filice, Giovanni Da San Martino, and Alessandro Moschitti. 2017. KeLP at SemEval-2017 task 3: Learning pairwise patterns in community question answering. In *SemEval-2017*. 326–333.

[14] Bent Fuglede and Flemming Topsoe. 2004. Jensen-Shannon divergence and Hilbert space embedding. In *ISIT*. IEEE, 31.

[15] Yichen Gong, Heng Luo, and Jian Zhang. 2017. Natural language inference over interaction space. *arXiv* (2017).

[16] Sparsh Gupta and Vitor R Carvalho. 2019. FAQ Retrieval Using Attentive Matching. In *SIGIR*. ACM, 929–932.

[17] Dan Gusfield. 1997. *Algorithms on strings, trees, and sequences: computer science and computational biology*.

[18] Francisco Guzmán, Lluís Màrquez, and Preslav Nakov. 2019. Machine translation evaluation meets community question answering. *arXiv* (2019).

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *CVPR*. 770–778.

[20] Doris Hoogeveen, Andrew Bennett, Yitong Li, Karin M Verspoor, and Timothy Baldwin. 2018. Detecting misflagged duplicate questions in community question-answering archives. In *AAAI*.

[21] Doris Hoogeveen, Karin M Verspoor, and Timothy Baldwin. [n.d.]. CQADupStack: A benchmark data set for community question-answering research. In *ADCS*.

[22] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. 2017. Densely connected convolutional networks. In *CVPR*. 4700–4708.

[23] Paul Jaccard. 1901. Étude comparative de la distribution florale dans une portion des Alpes et des Jura. *Bull Soc Vaudoise Sci Nat* 37 (1901), 547–579.

[24] Sarthak Jain and Byron C. Wallace. 2019. Attention is not Explanation. arXiv:cs.CL/1902.10186

[25] Jiwoon Jeon, W Bruce Croft, and Joon Ho Lee. 2005. Finding similar questions in large question and answer archives. In *CIKM*. ACM, 84–90.

[26] Zongcheng Ji, Fei Xu, Bin Wang, and Ben He. 2012. Question-answer topic model for question retrieval in community question answering. In *CIKM*. ACM.

[27] Di Liang, Fubao Zhang, Weidong Zhang, Qi Zhang, Jinlan Fu, Minlong Peng, Tao Gui, and Xuanjing Huang. 2019. Adaptive Multi-Attention Network Incorporating Answer Information for Duplicate Question Detection. (2019).

[28] Min Lin, Qiang Chen, and Shuicheng Yan. 2013. Network in network. *arXiv* (2013).

[29] Liyuan Liu, Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Jiawei Han. 2019. On the variance of the adaptive learning rate and beyond. *arXiv* (2019).

[30] Alessandro Moschitti. 2006. Efficient convolution kernels for dependency and constituent syntactic trees. In *ECML*. Springer, 318–329.

[31] Preslav Nakov, Lluís Màrquez, and Francisco Guzmán. 2016. It takes three to tango: triangulation approach to answer ranking in community question answering. In *EMNLP*. 1586–1597.

[32] Liang Pang, Yanyan Lan, Jiafeng Guo, Jun Xu, Shengxian Wan, and Xueqi Cheng. 2016. Text matching as image recognition. In *AAAI*.

[33] Xipeng Qiu and Xuanjing Huang. 2015. Convolutional neural tensor network architecture for community-based question answering. In *IJCAI*.

[34] Stephen E Robertson, Steve Walker, Susan Jones, Micheline M Hancock-Beaulieu, Mike Gatford, et al. 1995. Okapi at TREC-3. *NIST SP* 109 (1995), 109.

[35] Lorenza Romano, Milen Kouylekov, Idan Szpektor, Ido Dagan, and Alberto Lavelli. 2006. Investigating a generic paraphrase-based approach for relation extraction. In *EACL*.

[36] Wataru Sakata, Tomohide Shibata, Ribeka Tanaka, and Sadao Kurohashi. 2019. FAQ Retrieval using Query-Question Similarity and BERT-Based Query-Answer Relevance. *SIGIR* (2019).

[37] Keiji Shinzato, Tomohide Shibata, Daisuke Kawahara, and Sadao Kurohashi. 2012. Tsubaki: An open search engine infrastructure for developing information access methodology. *Journal of information processing* 20, 1 (2012), 216–227.

[38] Yi Tay, Luu Anh Tuan, and Siu Cheung Hui. 2018. Co-stack residual affinity networks with multi-level attention refinement for matching text sequences. *arXiv* (2018).

[39] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NIPS*. 5998–6008.

[40] Shengxian Wan, Yanyan Lan, Jiafeng Guo, Jun Xu, Liang Pang, and Xueqi Cheng. 2016. A deep architecture for semantic matching with multiple positional sentence representations. In *AAAI*.

[41] Shengxian Wan, Yanyan Lan, Jun Xu, Jiafeng Guo, Liang Pang, and Xueqi Cheng. 2016. Match-srnn: Modeling the recursive matching structure with spatial rnn. *IJCAI* (2016).

[42] Mengqiu Wang, Noah A Smith, and Teruko Mitamura. 2007. What is the Jeopardy model? A quasi-synchronous grammar for QA. In *EMNLP-CoNLL*. 22–32.

[43] Zhiguo Wang, Wael Hamza, and Radu Florian. 2017. Bilateral multi-perspective matching for natural language sentences. *arXiv* (2017).

[44] Guoshun Wu, Yixuan Sheng, Man Lan, and Yuanbin Wu. 2017. ECNU at SemEval-2017 Task 3: Using Traditional and Deep Learning Methods to Address Community Question Answering Task. In *SemEval-2017*. 365–369.

[45] Wei Wu, Xu Sun, and Houfeng Wang. 2018. Question condensing networks for answer selection in community question answering. In *ACL*. 1746–1755.

[46] Xiaobing Xue, Jiwoon Jeon, and W Bruce Croft. 2008. Retrieval models for question and answer archives. In *SIGIR*. ACM, 475–482.

[47] Liu Yang, Minghui Qiu, Swapna Gottipati, Feida Zhu, Jing Jiang, Huiping Sun, and Zhong Chen. 2013. Cqarank: jointly model topics and expertise in community question answering. In *CIKM*. ACM, 99–108.

[48] Runqi Yang, Jianhai Zhang, Xing Gao, Feng Ji, and Haiqing Chen. 2019. Simple and Effective Text Matching with Richer Alignment Features. *ACL* (2019).

[49] Alexander Yeh. 2000. More accurate tests for the statistical significance of result differences. In *COLING*. ACL, 947–953.

[50] Chengxiang Zhai and John Lafferty. 2004. A study of smoothing methods for language models applied to information retrieval. *TOIS* 22, 2 (2004), 179–214.

[51] Kai Zhang, Wei Wu, Haocheng Wu, Zhoujun Li, and Ming Zhou. 2014. Question retrieval with high quality answers in community question answering. In *CIKM*.

[52] Guangyou Zhou, Li Cai, Jun Zhao, and Kang Liu. 2011. Phrase-based translation model for question retrieval in community question answer archives. In *ACL*. ACL, 653–662.

[53] Guangyou Zhou, Yubo Chen, Daojian Zeng, and Jun Zhao. 2013. Towards faster and better retrieval models for question search. In *CIKM*. ACM, 2139–2148.

[54] Guangyou Zhou, Yang Liu, Fang Liu, Daojian Zeng, and Jun Zhao. 2013. Improving question retrieval in community question answering using world knowledge. In *IJCAI*.

[55] Tom Chao Zhou, Chin-Yew Lin, Irwin King, Michael R Lyu, Young-In Song, and Yunbo Cao. 2011. Learning to suggest questions in online forums. In *AAAI*.