Adaptive Multi-Attention Network Incorporating Answer Information for Duplicate Question Detection

Di Liang*, Fubao Zhang*, Weidong Zhang, Qi Zhang†, Jinlan Fu, Minlong Peng, Tao Gui and Xuanjing Huang
School of Computer Science, Shanghai Key Laboratory of Intelligent Information Processing
Fudan University, Shanghai, P.R.China 201203
{liangd17,fbzhang17,zhangyd17,qz,fujl16,mlpeng16,tgui16,xjhuang}@fudan.edu.cn

ABSTRACT

Community-based question answering (CQA), which provides a platform for people with diverse backgrounds to share information and knowledge, has become increasingly popular. With the accumulation of site data, methods to detect duplicate questions in CQA sites have attracted considerable attention. Existing methods typically use only questions to complete the task. However, the paired answers may also provide valuable information. In this paper, we propose an answer information- enhanced adaptive multi-attention network (AMAN) to perform this task. AMAN takes full advantage of the semantic information in the paired answers while alleviating the noise problem caused by adding the answers. To evaluate the proposed method, we use a CQADupStack set and the Quora question-pair dataset expanded with paired answers. Experimental results demonstrate that the proposed model can achieve state-of-the-art performance on the above two data sets.

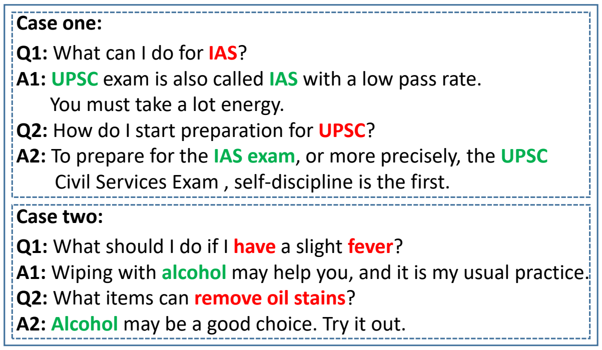
CCS CONCEPTS

• **Information systems → Near-duplicate and plagiarism detection**; *Question answering*.

KEYWORDS

duplicate question detection , adaptive multi-attention, community-based question answering

ACM Reference Format:
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. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '19), July 21–25, 2019, Paris, France.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3331184.3331228



Case one:
Q1: What can I do for **IAS**?
A1: **UPSC** exam is also called **IAS** with a low pass rate. You must take a lot energy.
Q2: How do I start preparation for **UPSC**?
A2: To prepare for the **IAS exam**, or more precisely, the **UPSC** Civil Services Exam , self-discipline is the first.

Case two:
Q1: What should I do if I **have** a slight **fever**?
A1: Wiping with **alcohol** may help you, and it is my usual practice.
Q2: What items can **remove oil stains**?
A2: **Alcohol** may be a good choice. Try it out.

Figure 1: Two examples from Quora. In case one, the corresponding answers explain that the IAS is equivalent to UPSC, which is crucial for determining the relationship between the both questions. In case two, the questions have distinct semantics, but their paired answers are semantically similar.

1 INTRODUCTION

Community-based question answering (CQA) websites such as Quora and Stack Overflow have grown in popularity in recent years. However, with the increase of the CQA archives, massive amounts of duplicate questions have accumulated. A large number of redundant questions make the maintenance for these sites harder and seriously affect the user experience. Therefore, it has become increasingly important to detect duplicate questions. There are two application scenarios for this technique. The first application scenario is used as a basic technique for CQA retrieval to judge whether one queried question is semantically equal to one historical question [29, 41, 43]. The other scenario is that a CQA forum needs to judge whether two historical questions are duplicates and then merge the duplicate historical questions on the site [12, 39, 42]. With an automatic detection method, the forum can organize questions and answers more efficiently. In this paper, we present a robust approach to the latter.

Question duplication is a pervasive issue in CQA, and existing works have studied various aspects of the detection problem. The study in [39] uses a distributed index and MapReduce framework to calculate pairwise similarity and to identify redundant data quickly and in a scalable manner. Zhang et al. [43] compute four similarity scores by comparing their titles, descriptions, latent topics, and tags of each pair of questions to detect duplicate posts in Stack Overflow.

[*] Equal contribution. Alphabetical order of the last name.
[†] Corresponding author.

SIGIR '19, July 21–25, 2019, Paris, France

ACM ISBN 978-1-4503-6172-9/19/07…$15.00
https://doi.org/10.1145/3331184.3331228

Zhang et al. [41] leverage continuous word vectors from the deep learning literature, topic model features, and phrases pairs that co-occur frequently in duplicate questions mined using machine translation systems. Hoogeveen et al. [12] find that for misflagged duplicate detection, meta data features that capture user authority, question quality, and relational data between questions, outperform pure text-based methods. In general, there are two major problems in duplicate detection, namely the lexical gap and essential constituents matching. Distributed representation is an effective way to tackle the lexical gap problem. Researchers have designed various similarity features based on word embeddings [30], or acquired representations of questions via neural networks and then calculated their similarity [7, 19]. And two approaches are proposed to integrate FrameNet parsing with neural networks to achieve essential constituents matching in [42].

Despite the above research improves the performance of previous state-of-the-art methods, some issues still have not been well solved. Due to the relatively short text and lexical gap, in many cases, the questions do not provide sufficient information. As a result, all of these methods depending on only questions suffer from having insufficient information to determine the relationship between questions. However, answers to questions usually explain the corresponding question in detail, they can be seen as a complementary information resource. Case one in the Figure 1 shows an example from Quora. In this case, although Q1 and Q2 share many words in common, their key concepts (IAS in Q1 and UPSC in Q2) cannot be linked via the both questions. Without the knowledge that IAS is equivalent to UPSC, existing methods that use only the questions will fail to accomplish the task. Defining and labeling knowledge bases for these rapidly-growing CQA websites is impractical, as this would consume too much time and resources. Answers often provide crucial information for linking these seemingly different concepts in the questions. However, the information provided by the answers is not always beneficial. Similar paired answers can also introduce noise to the detection of semantically different questions. Case two in Figure 1 illustrates another example from Quora. Q1 (What should I do if I have a slight fever?) and Q2 (What can be used to remove oil stains?) have distinct semantics, and previous methods may accurately distinguish the relationship between the two. Yet the semantics of the both corresponding answers are very similar. In this case, the introduction of answer information introduces complications in identifying a solution. Hence, it is nontrivial to incorporate answer information into neural networks with respect to duplicate question detection in a reasonable way.

Answer可以帮助判断question是否是相似的，但是他也会引入额外的噪声信息。

In this paper, we propose a novel method to perform this task, called the adaptive multi-attention network (AMAN). This model integrates external knowledge from paired answers for duplicate identification and filters out the noise introduced by answers adaptively. To obtain multi-level textual features, we use the concatenation of word embedding, character embedding, and syntactical features as the representation. To incorporate answer information and capture the text relevance effectively , we utilize three heterogeneous attention mechanisms: **self-attention**, which facilitates modeling of the temporal interaction in a long sentence; **cross attention**, which captures the relevance between questions and the relevance between answers; and **adaptive co-attention**, which extracts valuable knowledge from the answers. In an adaptive co-attention block, question-guided attention and answer-guided attention are combined to capture the semantic interaction between a question and its paired answer. We propose a gated fusion module to adaptively fuse the answer-based features. Then, to alleviate the noise introduced by paired answers, we utilize a filtration gate module as a filter. An interaction layer enhances the collected local semantic information of questions and answers. Finally, predictions are calculated based on the similarity features extracted from the question-answer pairs.

To demonstrate the effectiveness of our model, we evaluate it on CQADupStack set and Quora question-pair dataset with expanded paired answers. The experimental results on these two data sets reveal that our method can achieve better performance than those of previous methods.

The main contributions of this work can be summarized as follows:

- We take into account the noise problems that may be introduced by adding paired answers and study ways to integrate answer information into neural network-based methods to perform a duplicate detection task.
- We propose a novel method that integrates information extracted from paired answers into neural attention model to complete duplicate detection and alleviates possible noise introduced by this answer information.
- The experimental results on two data sets demonstrate that our model can achieve significantly better performance than those of current state-of-the-art methods.

2 RELATED WORK

Question duplication is a pervasive issue in CQA, and a number of studies have looked into related problems, including text relevance and question retrieval.

2.1 Text Relevance

Two categories of neural network-based models have been developed for this problem. The first set of models is sentence encoding-based. The models are developed from Siamese architecture [2] and aim to find a fixed-length vector representation for each of two sentences. Using the variant concatenation of the two sentence vectors, a neural network classifier is then employed to decide the relationship between the two sentences. The sentence encoder is usually based on RNN, CNN, or a self-attention network [1, 22, 32]. Sentence vectors produced by sentence encoding-based models usually generalize for a wide range of tasks. However, this kind of method doesn't explore the lower-level semantic interaction between sentences.

The second set of models uses the cross-sentence feature or inter-sentence attention from one sentence to another, and is hence referred to as a matching-aggregation framework. Rocktäschel et al. [28] are the first to use the attention-based method to improve the performance of LSTM. Wang et al. [38] try to match words in different sentences with word-by-word attention. Wang et al. [37] propose a multiple-perspectives attention mechanism to modeling the semantic matching between two sentences, and achieved state-of-the-art results on several relevant semantic matching tasks. Cheng et al. [5] enhance the attention mechanism by a memory network. Munkhdalai and Yu [23] use a tree structure to improve

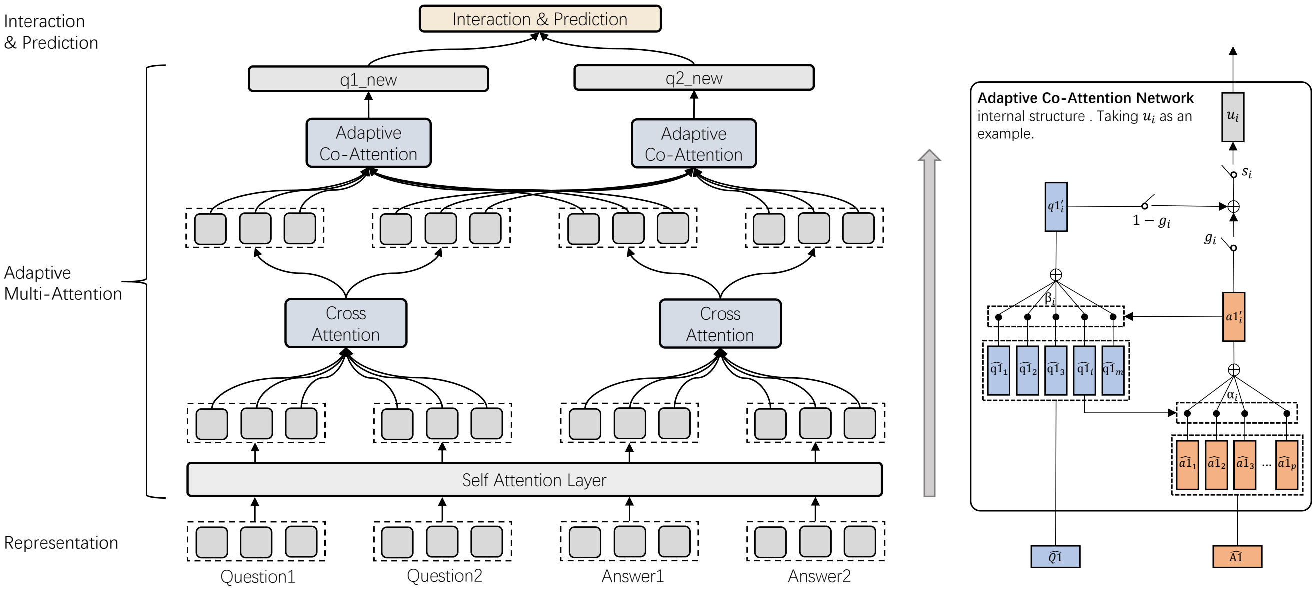


Figure 2: The overall view of our model. The left part is the main framework of this work. The right part is the detailed structure of the adaptive co-attention network. The thick arrow indicates that information flows from bottom to top.

the recurrent or recursive architecture for natural language inference and answer selection. Chen et al. [3] propose ESIM, which achieved state-of-the-art results on the SNLI dataset.

2.2 Question Retrieval

Most previous research has framed the CQA problem as a semantic matching task [9], and has relied on a number of extracted features to train models. Various supervised methods that use hand-crafted features or templates have been proposed for this task. In their paper, Wang et al. [36] tackle the similar question matching problem using syntactic parsing, while Zhou et al. [44] propose a phrase-based translation model for this task. Although these methods have shown impressive results, they are restricted in their modeling of word sequence information [27]. Recently, enabled by developments in distributed representation and deep learning methods, many word-embedding-based neural network models have been successfully used to tackle this problem from different angles [7, 23, 25, 30, 34]. Many researchers have also considered the use of different kinds of external resources. Zhou et al. [45] utilize semantic relations extracted from the global knowledge of Wikipedia.

In addition, some models for question retrieval are concerned about the important paired answer part [14, 31, 40]. But these studies ignore that the answer information may also introduce noise to the retrieval process. Different from the previous works, in this paper, we propose a novel method that integrates information extracted from paired answers into neural attention model to perform duplicate detection and to alleviate possible noise introduced by this answer information.

3 APPROACH

We define the duplicate question detection problem as a binary classification problem. Given two historical question-answer pairs (Q1-A1 and Q2-A2) on a CQA website, our goal is to judge whether the two historical questions are semantically equivalent or not. In this work, we propose an answer information-enhanced adaptive multi-attention network (AMAN) to incorporate external knowledge extracted from answers to complete this task. The overall architecture of the model is illustrated on the left part of Figure 2.

Our sentence matching architecture, AMAN , is composed of the following three components: (1) information representation layer, (2) adaptive multi-attention layer, and (3) interaction and prediction layer. The information representation layer combines the multi-level features as the question and answer representation. The adaptive multi-attention layer extracts the semantic connections between the questions and the paired answers. The interaction and prediction layer is designed to fuse local information for making a global decision at the sentence level.

3.1 Information Representation Layer

The information representation layer converts each word or phrase in the question-answer pairs into a vector representation and constructs the representation matrix for the sentences. We combine the multi-level features as the question and answer representation. Each token is represented as a vector by using the pre-trained word embedding such as GloVe [26], word2Vec [21], and fasttext [15]. It can also utilize the preprocessing tool, e.g. part-of-speech recognizer, named entity recognizer, lexical parser etc., to incorporate more syntactical and lexical information into the feature vector.

For AMAN, we use a concatenation of word embedding, character embedding, and syntactical features as the sentence representation. The word embedding is obtained by mapping token to high dimensional vector space by pre-trained word vector (300D Glove 840B), and the word embedding is updated during training. Character-level embedding could alleviate out-of-vocabulary (OOV) problems and capture helpful morphological information. As in [16, 18], we filter the character embedding with 1D convolution kernel. The character convolutional feature maps are then max pooled over the time dimension for each token to obtain a vector.

As in [4], the syntactical features consist of one-hot part-of-speech (POS) tagging feature and binary exact match (EM) feature. For one question or answer, the EM value is activated if the same word is found in the other question or answer.

Next, AMAN adopts bidirectional Long Short-Term Memory network (Bi-LSTM) [10] to model the internal temporal interaction on both directions of questions and answers. Consider two question-answer pairs (Q1-A1 and Q2-A2), we have got their multi-level features representation. Suppose the length of Q1, Q2, A1, and A2 are m, n, p, and l, respectively. These multi-level features representation are then passed to a Bi-LSTM encoder to obtain the context-dependent hidden state matrix, i.e, $\mathbf{Q1} = \{\mathbf{q1}_i | \mathbf{q1}_i \in \mathbb{R}^d, i = 1, 2, ..., m\}$, $\mathbf{Q2} = \{\mathbf{q2}_i | \mathbf{q2}_i \in \mathbb{R}^d, i = 1, 2, ..., n\}$, $\mathbf{A1} = \{\mathbf{a1}_i | \mathbf{a1}_i \in \mathbb{R}^d, i = 1, 2, ..., p\}$, and $\mathbf{A2} = \{\mathbf{a2}_i | \mathbf{a2}_i \in \mathbb{R}^d, i = 1, 2, ..., l\}$, where d is the dimension of Bi-LSTM's hidden state.

3.2 Adaptive Multi-attention Layer

Modeling local semantic information for words and their context is the basic procedure for determining the semantic relation between sentences. Generally in neural network methods, this procedure is achieved with some forms of soft alignment. Answers to questions usually explain the corresponding question in detail, they can be seen as a complementary information resource. However, as shown in Figure 1, the introduction of answer information sometimes may also have a negative impact on the detection. In this layer , we utilize three heterogeneous attention mechanisms to incorporate answer information into question pair matching and adaptively filter out the noise introduced by adding paired answers.

3.2.1 *Self-attention.*

In order to further model the temporal interaction between words and tackle the long-term dependency in a long sentence, we additionally introduce the self-attention mechanism. Formally, for question one, we first compute a self-attention matrix $\mathbf{E} \in \mathbb{R}^{m \times m}$:

$$\mathbf{E}_{i,j} = \langle \mathbf{q1}_i, \mathbf{q1}_j \rangle, \tag{1}$$

where $\langle \cdot, \cdot \rangle$ denotes the inner production operation. $\mathbf{E}_{i,j}$ indicates the relevance between the i-th word and j-th word in question one.

Then the self attentive vector for each word can be computed as follow:

$$\mathbf{e}_{q1_i} = softmax(\mathbf{E}_{i,:}), \tag{2}$$

$$\overline{\mathbf{q1}}_i = \mathbf{Q1} \cdot \mathbf{e}_{q1_i}, \tag{3}$$

We can similarly derive the self attentive vector for question two, answer one, and answer two as $\overline{\mathbf{q2}}_i$, $\overline{\mathbf{a1}}_j$, and $\overline{\mathbf{a2}}_k$, respectively.

3.2.2 *Cross attention.*

Cross attention captures the relevance between both questions and between both answers. For the both questions, we first compute a co-attention matrix $\mathbf{C} \in \mathbb{R}^{m \times n}$. Each element $\mathbf{C}_{i,j} \in R$ indicates the relevance between the i-th word of question one and the j-th word of question two. Formally, the co-attention matrix could be computed as:

$$\mathbf{C}_{i,j} = \mathbf{v}^T tanh(\mathbf{W}(\overline{\mathbf{q1}}_i \odot \overline{\mathbf{q2}}_j)), \tag{4}$$

where $\mathbf{W} \in \mathbb{R}^{k \times d}$, $\mathbf{v} \in \mathbb{R}^k$, and $\odot$ denotes the element-wise production operation. Then the attentive matrix for question one and question two could be formalized as $\widehat{\mathbf{Q1}}$ and $\widehat{\mathbf{Q2}}$:

$$\mathbf{c}_{\widehat{q1}_i} = softmax(\mathbf{C}_{i,:}), \quad \mathbf{c}_{\widehat{q2}_j} = softmax(\mathbf{C}_{:,j}), \tag{5}$$

$$\widehat{\mathbf{q1}}_i = \overline{\mathbf{Q2}} \cdot \mathbf{c}_{\widehat{q1}_i}, \quad \widehat{\mathbf{q2}}_j = \overline{\mathbf{Q1}} \cdot \mathbf{c}_{\widehat{q2}_j}, \tag{6}$$

$$\widehat{\mathbf{Q1}} = (\widehat{\mathbf{q1}}_1, \widehat{\mathbf{q1}}_2, ..., \widehat{\mathbf{q1}}_m), \tag{7}$$

$$\widehat{\mathbf{Q2}} = (\widehat{\mathbf{q2}}_1, \widehat{\mathbf{q2}}_2, ..., \widehat{\mathbf{q2}}_n), \tag{8}$$

where $\overline{\mathbf{Q1}} = (\overline{\mathbf{q1}}_1, \overline{\mathbf{q1}}_2, ..., \overline{\mathbf{q1}}_m)$ and $\overline{\mathbf{Q2}} = (\overline{\mathbf{q2}}_1, \overline{\mathbf{q2}}_2, ..., \overline{\mathbf{q2}}_n)$. In a similar way, we get the attentive matrix for the corresponding answers:

$$\widehat{\mathbf{A1}} = (\widehat{\mathbf{a1}}_1, \widehat{\mathbf{a1}}_2, ..., \widehat{\mathbf{a1}}_p), \tag{9}$$

$$\widehat{\mathbf{A2}} = (\widehat{\mathbf{a2}}_1, \widehat{\mathbf{a2}}_2, ..., \widehat{\mathbf{a2}}_l), \tag{10}$$

3.2.3 *Adaptive Co-attention.*

Inspired by previous works [20], adaptive co-attention includes question-guided attention and answer-guided attention to capture the semantic interaction between a question and its paired answer. Taking the time step i as an example, the internal structure of adaptive co-attention is shown on the right part of Figure 2. We propose the use of a gated fusion module to fuse the features adaptively. Then, to reduce the possibility of noise introduced by paired answer information, we utilize the filtration gate to adaptively filter out some of the useless answer information.

Formally, after the cross attention layer, $\widehat{\mathbf{q1}}_i$ is i-th word feature of the question one, and $\widehat{\mathbf{A1}}$ is the answer one feature matrix. We feed these through a single layer neural network followed by a softmax function to generate the attention distribution over the answer one:

$$\mathbf{z}_i = tanh(\mathbf{W}_{\widehat{A1}} \widehat{\mathbf{A1}} \oplus (\mathbf{W}_{\widehat{q1}_i} \widehat{\mathbf{q1}}_i + \mathbf{b}_{\widehat{q1}_i})), \tag{11}$$

$$\boldsymbol{\alpha}_i = softmax(\mathbf{W}_{\alpha_i} \mathbf{z}_i + \mathbf{b}_{\alpha_i}), \tag{12}$$

where $\mathbf{W}_{\widehat{A1}}$, $\mathbf{W}_{\widehat{q1}_i}$, $\mathbf{W}_{\alpha_i}$, $\mathbf{b}_{\widehat{q1}_i}$, and $\mathbf{b}_{\alpha_i}$ are parameters, and $\mathbf{W}_{\widehat{A1}}$, $\mathbf{W}_{\widehat{q1}_i} \in \mathbb{R}^{k \times d}$ and $\mathbf{W}_{\alpha_i} \in \mathbb{R}^{1 \times 2k}$. In addition, we use $\oplus$ to denote the concatenation of the answer one feature matrix and word feature vector of the question one. The concatenation between a matrix and a vector is performed by concatenating each column of the matrix by the vector.

Based on the attention distribution $\boldsymbol{\alpha}_i$, which is the weight corresponding to each word of the answer one, the new answer one vector related to i-th word in the question one can be obtained by:

$$\mathbf{a1}'_i = \widehat{\mathbf{A1}} \cdot \boldsymbol{\alpha}_i, \tag{13}$$

Next, we use the new answer one vector $\boldsymbol{a1}'_i$ to conduct the answer-based attention of the question one.

$$\widehat{\mathbf{z}}_i = tanh(\mathbf{W}_{\widehat{Q1}}\widehat{\mathbf{Q1}} \oplus (\mathbf{W}_{\widehat{Q1},a1'_i}\mathbf{a1}'_i + \mathbf{b}_{\widehat{Q1},a1'_i})), \quad (14)$$

$$\boldsymbol{\beta}_i = softmax(\mathbf{W}_{\beta_i}\widehat{\mathbf{z}}_i + \mathbf{b}_{\beta_i}), \quad (15)$$

Then, we acquire a new representation, $\mathbf{q1}'_i$:

$$\mathbf{q1}'_i = \widehat{\mathbf{Q1}} \cdot \boldsymbol{\beta}_i, \quad (16)$$

where $\mathbf{W}_{\widehat{Q1}}, \mathbf{W}_{\widehat{Q1},a1'_i} \in \mathbb{R}^{k \times d}$, and $\mathbf{W}_{\beta_i} \in \mathbb{R}^{1 \times 2k}$. We propose a gated fusion to fuse question feature and answer feature:

$$\mathbf{a1}''_i = tanh(\mathbf{W}_{a1'_i}\mathbf{a1}'_i + \mathbf{b}_{a1'_i}), \quad (17)$$

$$\mathbf{q1}''_i = tanh(\mathbf{W}_{q1'_i}\mathbf{q1}'_i + \mathbf{b}_{q1'_i}), \quad (18)$$

$$\mathbf{g}_i = \sigma(\mathbf{W}_{g_i}(\mathbf{a1}''_i \oplus \mathbf{q1}''_i)), \quad (19)$$

$$\boldsymbol{v}_i = \mathbf{g}_i\mathbf{a1}''_i + (1 - \mathbf{g}_i)\mathbf{q1}''_i, \quad (20)$$

where σ is the logistic sigmoid activation, $\mathbf{g}_i$ is the gate applied to the new answer vector $\mathbf{a1}''_i$, and $\boldsymbol{v}_i$ is the fusion feature that incorporates the question information and its paired answer information.

Because the fusion feature contains answer information, and it may introduce some noise, we use a filtration gate to combine the fusion feature and the original feature. The filtration gate is a scalar in the range of $[0, 1]$. When the fusion feature is helpful to improve the performance , the filtration gate is 1; otherwise, the value of the filtration gate is 0. The filtration gate $\mathbf{s}_i$ and the answer-information-enhanced feature $\widetilde{\mathbf{q1}}_i$ of question one are defined as follows:

$$\mathbf{s}_i = \sigma(\mathbf{W}_{s_i,\widehat{q1}_i}\widehat{\mathbf{q1}}_i \oplus (\mathbf{W}_{v_i,s_i}\boldsymbol{v}_i + \mathbf{b}_{v_i,s_i})), \quad (21)$$

$$\boldsymbol{u}_i = \mathbf{s}_i(tanh(\mathbf{W}_{v_i}\boldsymbol{v}_i + \mathbf{b}_{v_i})), \quad (22)$$

$$\widetilde{\mathbf{q1}}_i = \mathbf{W}_{\widetilde{q1}_i}(\widehat{\mathbf{q1}}_i \oplus \boldsymbol{u}_i), \quad (23)$$

where $\mathbf{W}_{s_i,\widehat{q1}_i}, \mathbf{W}_{v_i,s_i}, \mathbf{W}_{v_i}, \mathbf{W}_{\widetilde{q1}_i}, \mathbf{b}_{v_i,s_i}$, and $\mathbf{b}_{v_i}$ are parameters, $\boldsymbol{u}_i$ is the reserved features after filtration gate filter out noise.

We can similarly derive the answer-information-enhanced vector for question two as $\widetilde{\mathbf{q2}}_j$.

3.3 Interaction and Prediction Layer

Inspired by previous works[3, 22], we further enhance the collected local semantic information by combining the question representation and corresponding answer-information-enhanced vector of question. More formally:

$$\boxed{\mathbf{q1}^m_i} = [\mathbf{q1}_i; \widetilde{\mathbf{q1}}_i; \mathbf{q1}_i - \widetilde{\mathbf{q1}}_i; \mathbf{q1}_i \odot \widetilde{\mathbf{q1}}_i], \quad (24)$$

$$\boxed{\mathbf{q2}^m_j} = [\mathbf{q2}_j; \widetilde{\mathbf{q2}}_j; \mathbf{q2}_j - \widetilde{\mathbf{q2}}_j; \mathbf{q2}_j \odot \widetilde{\mathbf{q2}}_j], \quad (25)$$

where $[\cdot;\cdot;\cdot;\cdot]$ refers to the concatenation operation. In the formula, we first calculate the difference and the element-wise product for $(\mathbf{q1}_i, \widetilde{\mathbf{q1}}_i)$ as well as for $(\mathbf{q2}_j, \widetilde{\mathbf{q2}}_j)$.

Then, BiLSTMs are trained to learn to modeling vectors which contain the crucial information for judging the relationship between two sentences:

$$\boxed{\mathbf{q1}^v_i} = BiLSTM(\mathbf{q1}^m_i, \mathbf{q1}^v_{i-1}, \mathbf{q1}^v_{i+1}), \quad (26)$$

$$\boxed{\mathbf{q2}^v_j} = BiLSTM(\mathbf{q2}^m_j, \mathbf{q2}^v_{j-1}, \mathbf{q2}^v_{j+1}), \quad (27)$$

Table 1: CQADupStack sub-forum statistics.

Sub-forum	pairs	of duplicates
android	1,866	622
english	5,076	1,692
gaming	3,531	1,177
gis	978	326
mathematica	1,302	434
physics	2,196	732
programmers	2,637	879
stats	645	215
tex	4,560	1,520
unix	2,466	822
webmasters	1,899	633
wordpress	864	282
all	**28,020**	**9,334**

Our model converts the resulting vectors obtained above to a fixed-length vector with pooling and feeds it to the final classifier to determine the overall relationship. More specifically, we compute max pooling and mean pooling for $\mathbf{Q1}^v$ and $\mathbf{Q2}^v$. where $\mathbf{Q1}^v = (\mathbf{q1}^v_1, \mathbf{q1}^v_2, ..., \mathbf{q1}^v_m)$ and $\mathbf{Q2}^v = (\mathbf{q2}^v_1, \mathbf{q2}^v_2, ..., \mathbf{q2}^v_n)$. All these vectors are then concatenated into a fixed-length vector $\mathbf{r}$. Formally:

$$\mathbf{r}^{mean}_{Q1} = \sum_{i=1}^{m} \frac{\mathbf{q1}^v_i}{m}, \quad \mathbf{r}^{max}_{Q1} = \max_{i=1}^{m} \mathbf{q1}^v_i, \quad (28)$$

$$\mathbf{r}^{mean}_{Q2} = \sum_{j=1}^{n} \frac{\mathbf{q2}^v_j}{n}, \quad \mathbf{r}^{max}_{Q2} = \max_{j=1}^{n} \mathbf{q2}^v_j, \quad (29)$$

$$\mathbf{r} = [\mathbf{r}^{mean}_{Q1}; \mathbf{r}^{max}_{Q1}; \mathbf{r}^{mean}_{Q2}; \mathbf{r}^{max}_{Q2}], \quad (30)$$

We then put the obtained final global representation $\boldsymbol{r}$ into our prediction layer to determine whether Q1 and Q2 are semantically equivalent.

The duplicate question detection task requires the model to predict whether the given question pair $(Q1, Q2)$ is semantically identical or not, hence it is a binary classification task. We use a multi-layer perceptron (MLP) classifier to predict the label:

$$\boldsymbol{v} = ReLU(\mathbf{W}_r\mathbf{r} + \mathbf{b}_r), \quad (31)$$

$$\hat{\boldsymbol{y}} = softmax(\mathbf{W}_v\boldsymbol{v} + \mathbf{b}_v). \quad (32)$$

where $\mathbf{W}_r, \mathbf{b}_r, \mathbf{W}_v$, and $\mathbf{b}_v$ are trainable parameters. The entire model is trained end-to-end, optimizing the standard binary cross-entropy loss function.

4 EXPERIMENT

In this section, we present the evaluation of our model. We first perform quantitative evaluation, comparing our model with other competitive models. We then conduct some qualitative analyses to understand the ability of AMAN to incorporate answer information and adaptively filter out noise.

Table 2: Hyper-parameters configuration.

Hyper-parameters	Value
Word embedding size	$d_e = 300$
character embedding size	$d_c = 100$
convolution kernel size	$d_k = 5$
Initial learning rate	$\alpha = 0.001$
Adam β_1	$\beta_1 = 0.9$
Adam β_2	$\beta_2 = 0.999$
Dropout rate	$p = 0.2$
Batch size	$b = 64$
LSTM hidden size	$d_{h^x} = 300$
MLP hidden size	$d_{h^t} = 300$

Table 3: Overall results on AeQQP.

Model	Accuracy
InferSent [6]	84.00
SSE [24]	86.62
PWIM [11]	72.59
Multi-Perspective-CNN [37]	78.98
Multi-Perspective-LSTM [37]	79.12
BiMPM [37]	87.32
pt-DECATT [35]	86.43
ESIM [3]	84.35
AF-DMN [8]	87.61
DIIN [4]	88.20
AMAN(ours)	**90.07**

4.1 Dataset

In this work, we introduce two datasets to evaluate our model. In addition to CQADupStack, we expand the Quora Question Pairs (QQP) dataset with the paired answers and named it the answer-enhanced QQP (AeQQP).

AeQQP : Each sample in the QQP dataset contains two questions and is annotated with a binary label indicating whether these two questions are semantically equivalent. The question pairs in the dataset are not restricted to any subject. In the original dataset, the answers to the questions are not contained. To evaluate the effectiveness of our model, we collect the answers from Quora for question pairs in the QQP dataset. More specifically, in the QQP dataset, there are a total of 404,302 question pairs formed by 537,933 distinct questions. We crawl the answer recommended by Quora (usually the answer with the most *upvotes*) for each question in the dataset. In total, we get 290,391 question pairs where both questions were answered. We construct the AeQQP dataset with the $290k$ question pairs with both questions answered and their corresponding answers. We split the dataset into three parts: training set, development set, and testing set, which contain $270k$, $10k$, and $10k$ question-answer pairs, respectively. Accuracy is used as the evaluation metric on this dataset.

CQADupStack: This is a benchmark dataset for use in community question-answering (CQA) research [13]. It contains threads from twelve StackExchange sub-forums, annotated with duplicate question information. Table 1 gives the total number of the question pairs with both questions answered and duplicate questions for the twelve sub-forums. The training, development, and test split follows a ratio of 8:1:1. For CQADupStack, Precision , Recall, $F1$ score, and Accuracy are used as the evaluation metrics in this work. $F1$ score is the harmonic mean of Precision and Recall, wherein Recall reflects the ability to identify duplicate pairs among the true duplicate pairs.

4.2 Models for Comparing

To analyze the effectiveness of our model, we evaluate some traditional and state-of-the-art methods as baselines as follows on the above two data sets:

- **InferSent** [6] is a sentence encoding-based model. InferSent adopts Bi-LSTM max-pooling sentence encoder and passes the independent vector representations of two questions through an MLP classifier to make the final prediction.
- **SSE** [24] is a simple sequential sentence encoder for multi-domain natural language inference and is based on stacked bidirectional LSTM-RNNs with shortcut connections and fine-tuning of word embeddings. It enhances multi-layer Bi-LSTM with a skip connection.
- **PWIM** [11] uses cosine similarity, Euclidean distance, and dot product to calculate the word-pair interactions.
- **pt-DECATT** [35] is variant of decomposable attention models based on word-level embedding and character-level n-gram embedding.
- **ESIM** [3] is a previous state-of-the-art model for the natural language inference (NLI) task. It is a sequential model that incorporates the chain LSTM and the tree LSTM to infer local information between two sentences.
- **DIIN** [4] is a novel class of neural network architectures that is able to achieve high-level understanding of the sentence pair by hierarchically extracting semantic features from the interaction space. The model uses word-by-word dimension-wise alignment tensors to encode the high-order alignment relationship between sentence pairs.
- **AF-DMN** [8] stacks multiple computational blocks in its matching layer to learn the interaction of the sentence pair better.
- **Multi-Perspective-CNN** [37] changes the cosine similarity calculation layer with multi-perspective cosine matching function based on "Siamese-CNN" which implements the sentence encoder with a CNN.
- **Multi-Perspective-LSTM** [37] is an identical idea to the Multi-Perspective-CNN, but uses "Siamese-LSTM" instead of "Siamese-CNN".
- **BiMPM** is also proposed in [37]. The model combines the above two models. All these models employ a multi-perspective matching mechanism in sentence pair modeling tasks.

The first two models are both sentence encoding-based models, and all other models use some kind of cross sentence feature.

Table 4: Overall results on CQADupStack.

MODEL	Precision	Recall	F1	Accuracy
Multi-Perspective-CNN [37]	82.81	92.23	87.13	90.12
Multi-Perspective-LSTM [37]	83.64	94.08	87.98	90.15
BiMPM [37]	84.78	97.21	90.04	94.14
ESIM [3]	87.83	95.20	90.81	93.85
AF-DMN [8]	89.22	93.66	90.92	94.72
DIIN [4]	89.46	94.60	91.36	94.73
AMAN(ours)	**90.52**	**97.87**	**94.05**	**96.28**

Table 5: Sub-forum results of CQADupStack.

sub-forum	ESIM		DIIN		AMAN(ours)	
	F1	Accuracy	F1	Accuracy	F1	Accuracy
android	84.17	90.21	89.05	92.61	**89.95**	**93.44**
english	86.70	91.46	90.60	92.26	**91.97**	**94.00**
gaming	86.29	92.28	88.88	**93.64**	**90.80**	93.62
jgis	82.23	88.79	87.29	89.89	**90.08**	**92.21**
mathematica	81.58	87.11	90.94	91.96	**92.23**	**94.01**
physics	85.62	86.17	90.28	90.30	**91.96**	**90.56**
programmers	85.11	**91.56**	87.36	89.41	**89.36**	91.18
stats	85.36	85.90	86.90	91.01	**90.75**	**91.59**
tex	90.58	93.09	91.12	93.65	**94.45**	**95.23**
unix	86.22	90.75	88.34	91.50	**89.93**	**92.06**
webmasters	83.28	89.86	86.84	**91.01**	**90.02**	90.86
wordpress	85.11	86.00	89.29	90.14	**90.63**	**91.89**

4.3 Experiment Configurations

We adopt the hyper-parameters as shown in Table 2. In this work, an Adam [17] optimizer with β_1 as 0.9 and β_2 as 0.999 is used to optimize all trainable parameters. The initial learning rate is set to 0.001 and is halved when the accuracy on the dev set decreases. We use a batch size of 64. All hidden states of LSTMs and MLPs have 300 dimensions. We also apply dropout [33] on the word embeddings and all MLPs to avoid over-fitting, and the dropout rate is set to 0.2.

The length of all questions and answers is truncated to 40 and 100 respectively. For initialization, we initialize the word embeddings with a 300D Glove 840B [26], and the out-of- vocabulary (OOV) words are randomly initialized. All word embeddings are updated during training. Parameters, including neural network parameters and OOV word embeddings, are initialized with a uniform distribution between $[-0.01, 0.01]$. The character embeddings are randomly initialized with 100D. We crop or pad each token to have 16 characters. And the 1D convolution kernel size for character embedding is 5.

4.4 Quantitative Results

In this subsection, we compare our model performance to that of other neural network-based models on the AeQQP and CQADupStack dataset. On the both datasets, our model **AMAN** uses question-answer pairs information, while other compared models are only trained on the question pairs dataset.

As illustrated in Table 3, our model outperforms the baselines and achieves an accuracy of 90.07% in the test set of the AeQQP dataset. In Table 3, the first two models InferSent and SSE are both sentence encoding-based models, and all other compared models use some kind of cross-sentence feature.

Meanwhile, the results demonstrate that the models utilizing cross-sentence features achieve more competitive results in this task. This phenomenon shows that cross-sentence interaction operations, like cross attention, are crucial components for sentence modeling. We explore this idea in our multi-attention component to understand logical and semantic relationship between two sentences.

Precision, Recall, $F1$ score, and Accuracy are used as the evaluation metrics on the CQADupStack dataset. Table 4 shows the overall results of different models on the test set of the CQADupStack dataset. Our AMAN model achieves state-of-the-art performance on the all four evaluation metrics. In addition, we evaluate AMAN and two other strong baselines (ESIM and DIIN) on twelve sub-forum datasets separately. The models are trained and tested on every sub-forum dataset separately. Table 5 shows the performance of the different models on every sub-forum dataset. Our model achieves the best performance on the most of sub-forum sets.

The above results demonstrate that the answer information could be an essential knowledge source for duplicate question detection, and our model makes effective use of this information.

4.5 Answer Information Research

To further explore the impact of the answer information on duplicate question detection, we perform additional experiments on the AeQQP dataset.

As illustrated in Table 6, we train the ESIM on the answer pairs dataset of AeQQP train set, and the ESIM achieves an accuracy of 67.82% in the test set. The result demonstrates that this task can be done pretty well using only the answer information. This verifies the speculation that answers usually explain the corresponding questions in detail, and hence they could provide sufficient information for duplicate identification.

In addition, we concatenate the multi-level features representation of the question and the corresponding answer to acquire the Q-A representation. Then, we feed the the Q-A representation to the ESIM in the process of training and testing instead of only using the question representation. With this method, the performance drops sharply to 77.56% from 84.35%. The result illustrates that the corresponding answer information may also cause trouble for the solution of the task, and the naive way to incorporate answer information could introduce noise to detection.

Generally, the answer information for this task is usually a mixture of valuable information and other redundant information. Hence, how to incorporate answer information into neural networks for duplicate question detection should be investigated.

Table 6: Answer information research results.

Method	Accuracy
ESIM (Answer Pairs)	67.82
ESIM (Q-A) Pairs	77.56
ESIM (Question Pairs)	84.35
AMAN (ours)	**90.07**

Table 7: Ablation experiment results on AeQQP.

Method	Accuracy
1. AMAN (- self-att)	89.60
2. AMAN (- co-att)	88.42
3. AMAN (- adaptive co-att + co-att)	89.19
4. AMAN (- adaptive co-att)	87.58
5. AMAN (- interaction)	88.13
6. AMAN (- char-emb - syntactical fea)	89.10
7. **AMAN(ours)**	**90.07**

4.6 Ablation Study

We conduct an ablation study on our base model to examine the effectiveness of each component. We study our model on the AeQQP dataset. The experimental results are shown in Table 7.

First, we study how self-attention contributes to the system. After removing the self-attention component, we find that the performance degrades to 89.60% for test accuracy. Simple self-attention further models the temporal interaction between words and tackles the long-term dependency in a long sentence to acquire stronger representation. In the experiment 2, we remove the cross attention acting on between the both questions and between the both answers, and the performance drops to 88.42%. Cross attention can capture the relevance between both sentences, and the relevance information is crucial for this task. (-adaptive co-att + co-att) indicates that we introduce ordinary cross attention to integrate the paired answers information in instead of adaptive co-attention. Model performance decrease by nearly 0.9 percentage points. Furthermore, the result of experiment 4 shows that our AMAN, without adaptive co-attention component, declines to an accuracy of 87.58% in the test set, which is equal to the removal of the answers feature to only depend on the question information. The above two experiments reflect that our adaptive co-attention component integrates the answer information and alleviates the noise problem effectively. To show that the interaction layer can enhance the collected local semantic information and help to determine the overall relationship between both questions, we remove this component as a comparison in the experiment 5. The result of 88.13% demonstrates that our interaction component plays a crucial role in achieving competitive performance. In the last comparative experiment, we explore the role of multi-level features. We remove character embedding and syntactical features and just keep word embedding as the representation. The performance of the model is reduced to 89.10% on the test set.

In conclusion, due to the effective combination of each component, our model integrates valuable information from paired answers for duplicate identification and adaptively filters out the noise introduced by answers.

4.7 Case Study

To visually demonstrate the validity of the model, we do a qualitative study using the two cases in Figure 1. The qualitative results are demonstrated in Table 8. Only depending on question pairs information, the ESIM and DIIN are able to capture the distinct semantics of Q1 and Q2 in case 2, but they are unconcerned about the association between IAS and UPSC in case 1. Therefore the above two methods correctly judge the label of case 2, while determining the label of case 1 as NO. The ESIM(Q-A pairs) is trained and tested using the concatenation of the question representation and the corresponding answer representation. With the knowledge provided by the answer pair that IAS is equivalent to UPSC, the ESIM(Q-A pairs) makes a correct judgment in case 1. But because of the interference provided by similar answers, the model fails in case 2.

Our model AMAN makes the correct predictions in the both cases. With the filtration gate being set to 1 automatically, the proposed model AMAN incorporates the information extracted from the answer pair into the detection and links the key concepts (IAS in Q1 and UPSC in Q2) in case 1. With the filtration gate as 0 adaptively, the model filters out the noise introduced by the similar answers, and correctly judges that both questions are different in case 2.

4.8 Parameter Sensitivity

In this section, we evaluate the impact of the hidden state dimension of LSTMs and the answer sentence length on the AeQQP.

Table 8: Qualitative results. The ESIM(Q-A pairs) indicates that we feed the Q-A representation to the ESIM instead of only question representation. Our model is able to effectively utilize the information extracted from answer pairs and adaptively filter out the resulting noise.

		ESIM	ESIM(Q-A pairs)	DIIN	AMAN(ours)
Case 1	Q 1: What can I do for IAS? A 1: To prepare for IAS exam, or more precisely, the UPSC Civil Services Exam, self-discipline is the first. Q 2: How do I start preparation for UPSC? A 2: UPSC exam is also called IAS with a low pass rate. You must take a lot energy. Label: YES	prediction: NO	prediction: YES	prediction: NO	prediction: YES filtration gate: 1
Case 2	Q 1: What should I do if I have a slight fever? A 1: Wiping with alcohol may help you, and it is my usual practice. Q 2: What items can remove oil stains? A 2: Alcohol may be a good choice. Try it out. Label: NO	prediction: NO	prediction: YES	prediction: NO	prediction: NO filtration gate: 0

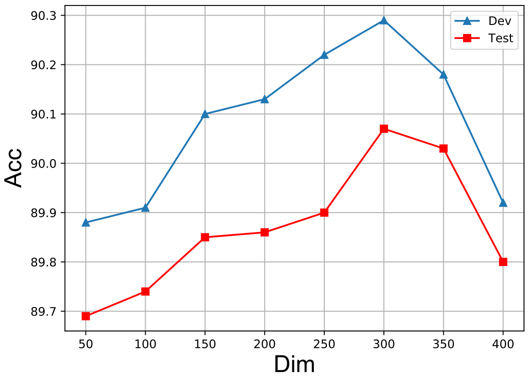


Figure 3: Results of our model influenced by different hidden state dimensions of LSTMs.

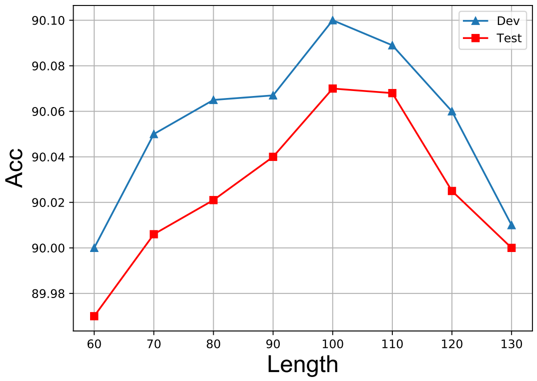


Figure 4: Results with answers that are truncated at different lengths.

First, we investigate the impact of different hidden state dimensions of LSTMs. Figure 3 shows our model's achieved results for different dimensions. As shown in the figure, when the hidden state size is less than 300, the performance of our model is increasing along with it. This trend indicates that a large hidden state size could enhance the performance of our model. When the dimension reaches 400, however, the performance drops on both the dev and test sets. This may be due to a requirement of more data for fitting such a large number of parameters. In our work, we get the best result when the hidden state dimensions of the LSTMs are set to 300.

We further compare the performance of our model with answers that are truncated at different lengths. As illustrated in Figure 4, our model achieve the best performance at the truncated length of answers as 100. As mentioned before, the answer information for this task is usually a mixture of valuable information and other redundant information. Therefore, a shorter truncation may cause the useful information to be lost, while a longer truncation may introduce more redundant information to aggravate the noise problem.

5 CONCLUSIONS

In this work, we show that the paired answers can provide effective information for duplicate question detection while they may simultaneously introduce noise to the detection. We propose an adaptive multi-attention network (AMAN), an effective method integrating external knowledge from answers for duplicate identification and filtering out the noise introduced by paired answers adaptively. This model consists of three layers: the *information representation layer* aims to obtain multi-level textual features as sentence representation; the *adaptive multi-attention layer* incorporates answer information into the neural attention model and captures the text relevance; and the *interaction and prediction layer* enhances the collected local semantic information of questions and answers to make a global decision at the sentence level. Experimental results on two data sets demonstrate that our model can achieve significantly better performance than those of current state-of-the-art methods.

ACKNOWLEDGMENTS

The authors wish to thank the anonymous reviewers for their helpful comments. This work was partially funded by China National Key RD Program (No.2018YFC0831105, 2017YFB1002104), National Natural Science Foundation of China (No.61751201, 61532011), Shanghai Municipal Science and Technology Major Project (No.2018SHZD ZX01), STCSM (No.16JC1420401, 17JC1420200), ZJLab.

REFERENCES

[1] Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015. A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326* (2015).

[2] Jane Bromley, Isäbelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. 1994. Signature verification using a" siamese" time delay neural network. In *Advances in neural information processing systems*. 737–744.

[3] Qian Chen, Xiaodan Zhu, Zhenhua Ling, Si Wei, Hui Jiang, and Diana Inkpen. 2016. Enhanced lstm for natural language inference. *arXiv preprint arXiv:1609.06038* (2016).

[4] Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Diana Inkpen, and Si Wei. 2018. Neural natural language inference models enhanced with external knowledge. In *ACL 2018 (Volume 1: Long Papers)*, Vol. 1. 2406–2417.

[5] Jianpeng Cheng, Li Dong, and Mirella Lapata. 2016. Long short-term memory-networks for machine reading. *arXiv preprint arXiv:1601.06733* (2016).

[6] Alexis Conneau, Douwe Kiela, Holger Schwenk, Loic Barrault, and Antoine Bordes. 2017. Supervised learning of universal sentence representations from natural language inference data. *arXiv preprint arXiv:1705.02364* (2017).

[7] Cícero Dos Santos, Luciano Barbosa, Dasha Bogdanova, and Bianca Zadrozny. 2015. Learning hybrid representations to retrieve semantically equivalent questions. In *ACL-IJCNLP 2015 (Volume 2: Short Papers)*, Vol. 2. 694–699.

[8] Chaoqun Duan, Lei Cui, Xinchi Chen, Furu Wei, Conghui Zhu, and Tiejun Zhao. 2018. Attention-Fused Deep Matching Network for Natural Language Inference.. In *IJCAI*. 4033–4040.

[9] Hanyin Fang, Fei Wu, Zhou Zhao, Xinyu Duan, Yueting Zhuang, and Martin Ester. 2016. Community-based question answering via heterogeneous social network learning. In *Thirtieth AAAI Conference on Artificial Intelligence*.

[10] Alex Graves and Jürgen Schmidhuber. 2005. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Networks* 18, 5-6 (2005), 602–610.

[11] Hua He and Jimmy Lin. 2016. Pairwise word interaction modeling with deep neural networks for semantic similarity measurement. In *NAACL 2016: Human Language Technologies*. 937–948.

[12] Doris Hoogeveen, Andrew Bennett, Yitong Li, Karin M Verspoor, and Timothy Baldwin. 2018. Detecting Misflagged Duplicate Questions in Community Question-Answering Archives. In *Twelfth International AAAI Conference on Web and Social Media*.

[13] Doris Hoogeveen, Karin M Verspoor, and Timothy Baldwin. 2015. CQADupStack: A benchmark data set for community question-answering research. In *Proceedings of the 20th Australasian Document Computing Symposium*. ACM, 3.

[14] Zongcheng Ji, Fei Xu, Bin Wang, and Ben He. 2012. Question-answer topic model for question retrieval in community question answering. In *Proceedings of the 21st ACM international conference on Information and knowledge management*. ACM, 2471–2474.

[15] Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759* (2016).

[16] Yoon Kim, Yacine Jernite, David Sontag, and Alexander M Rush. 2016. Character-Aware Neural Language Models.. In *AAAI*. 2741–2749.

[17] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[18] Jason Lee, Kyunghyun Cho, and Thomas Hofmann. 2016. Fully character-level neural machine translation without explicit segmentation. *arXiv preprint arXiv:1610.03017* (2016).

[19] Tao Lei, Hrishikesh Joshi, Regina Barzilay, Tommi Jaakkola, Katerina Tymoshenko, Alessandro Moschitti, and Lluis Marquez. 2016. Semi-supervised question retrieval with recurrent convolutions. *NAACL 2016* (2016).

[20] Jiasen Lu, Caiming Xiong, Devi Parikh, and Richard Socher. 2017. Knowing when to look: Adaptive attention via a visual sentinel for image captioning. In *CVPR*, Vol. 6. 2.

[21] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *NIPS*. 3111–3119.

[22] Lili Mou, Rui Men, Ge Li, Yan Xu, Lu Zhang, Rui Yan, and Zhi Jin. 2015. Natural language inference by tree-based convolution and heuristic matching. *arXiv preprint arXiv:1512.08422* (2015).

[23] Tsendsuren Munkhdalai and Hong Yu. 2017. Neural tree indexers for text understanding. In *Proceedings of the conference. Association for Computational Linguistics. Meeting*, Vol. 1. NIH Public Access, 11.

[24] Yixin Nie and Mohit Bansal. 2017. Shortcut-stacked sentence encoders for multi-domain inference. *arXiv preprint arXiv:1708.02312* (2017).

[25] Ankur P Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. 2016. A decomposable attention model for natural language inference. *arXiv preprint arXiv:1606.01933* (2016).

[26] Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*. 1532–1543.

[27] Xipeng Qiu and Xuanjing Huang. 2015. Convolutional Neural Tensor Network Architecture for Community-Based Question Answering.. In *IJCAI*. 1305–1311.

[28] Tim Rocktäschel, Edward Grefenstette, Karl Moritz Hermann, Tomáš Kočiskỳ, and Phil Blunsom. 2015. Reasoning about entailment with neural attention. *arXiv preprint arXiv:1509.06664* (2015).

[29] João António Rodrigues, Chakaveh Saedi, Vladislav Maraev, Joao Silva, and António Branco. 2017. Ways of asking and replying in duplicate question detection. In ** SEM 2017*. 262–270.

[30] MF Salvador, S Kar, T Solorio, and P Rosso. 2016. Combining lexical and semantic-based features for community question answering. *SemEval* (2016), 814–821.

[31] Anirban Sen, Manjira Sinha, and Sandya Mannarswamy. 2017. Improving Similar Question Retrieval using a Novel Tripartite Neural Network based Approach. In *Proceedings of the 9th Annual Meeting of the Forum for Information Retrieval Evaluation*. ACM, 1–5.

[32] Tao Shen, Tianyi Zhou, Guodong Long, Jing Jiang, Sen Wang, and Chengqi Zhang. 2018. Reinforced Self-Attention Network: a Hybrid of Hard and Soft Attention for Sequence Modeling. *arXiv preprint arXiv:1801.10296* (2018).

[33] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research* 15, 1 (2014), 1929–1958.

[34] Yi Tay, Luu Anh Tuan, and Siu Cheung Hui. 2017. A Compare-Propagate Architecture with Alignment Factorization for Natural Language Inference. *arXiv preprint arXiv:1801.00102* (2017).

[35] Gaurav Singh Tomar, Thyago Duque, Oscar Täckström, Jakob Uszkoreit, and Dipanjan Das. 2017. Neural paraphrase identification of questions with noisy pretraining. *arXiv preprint arXiv:1704.04565* (2017).

[36] Kai Wang, Zhaoyan Ming, and Tat-Seng Chua. 2009. A syntactic tree matching approach to finding similar questions in community-based qa services. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*. ACM, 187–194.

[37] Zhiguo Wang, Wael Hamza, and Radu Florian. 2017. Bilateral multi-perspective matching for natural language sentences. *arXiv preprint arXiv:1702.03814* (2017).

[38] Zhiguo Wang, Haitao Mi, and Abraham Ittycheriah. 2016. Sentence similarity learning by lexical decomposition and composition. *arXiv preprint arXiv:1602.07019* (2016).

[39] Yan Wu, Qi Zhang, and Xuanjing Huang. 2011. Efficient near-duplicate detection for q&a forum. In *IJCNLP*. 1001–1009.

[40] 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*. ACM, 371–380.

[41] Wei Emma Zhang, Quan Z Sheng, Jey Han Lau, and Ermyas Abebe. 2017. Detecting duplicate posts in programming QA communities via latent semantics and association rules. In *Proceedings of the 26th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 1221–1229.

[42] Xiaodong Zhang, Xu Sun, and Houfeng Wang. 2018. Duplicate Question Identification by Integrating FrameNet with Neural Networks. (2018).

[43] Yun Zhang, David Lo, Xin Xia, and Jian-Ling Sun. 2015. Multi-factor duplicate question detection in stack overflow. *Journal of Computer Science and Technology* 30, 5 (2015), 981–997.

[44] Guangyou Zhou, Li Cai, Jun Zhao, and Kang Liu. 2011. Phrase-based translation model for question retrieval in community question answer archives. In *ACL 2011: Human Language Technologies-Volume 1*. Association for Computational Linguistics, 653–662.

[45] Guangyou Zhou, Yang Liu, Fang Liu, Daojian Zeng, and Jun Zhao. 2013. Improving Question Retrieval in Community Question Answering Using World Knowledge. In *IJCAI*, Vol. 13. 2239–2245.