IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received March 6, 2020, accepted March 17, 2020, date of publication March 20, 2020, date of current version March 31, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2982268

Forum Duplicate Question Detection by Domain Adaptive Semantic Matching

ZHUOJIA XU AND HUA YUAN

Communication and Computer Network Laboratory of Guangdong, School of Computer Science and Engineering, South China University of Technology, Guangzhou 510641, China

Corresponding author: Zhuojia Xu (xuzhuojr@gmail.com)

This work was supported by the Graduate Education Innovation Program of Guangdong Province and the Guangdong Higher Education Teaching Reform Project (Research on Constructivism and its Application in the Teaching of Computer Networks) under Project 2015JGXM-ZD04.

ABSTRACT Community Question Answering (CQA) forums, such as Stack Overflow, Stack Exchange and Massive Open Online Course (MOOC) forums, spend a lot of manpower and time to manage duplicate questions on the forum. Mismatch of duplicate questions makes users keep asking ''new'' questions, and the continuous accumulation of duplicate questions may interfere with their information searching again, affecting user satisfaction. Neural Networks (NN) models for parsing semantics provide the possibility of end-to-end duplicate question detection. Whereas, due to lack of domain data and expertise, NN models for semantic parsing are rarely directly applied to CQA duplicate question detection. This paper proposes a Semantic Matching Model (SMM) integrated with the multi-task transfer learning framework for multi-domain forum duplicate question detection. By designing the word-to-sentence interaction mechanism based on the word-to-word interaction, SMM can automatically choose to ignore or pay attention to potential similar words according to the semantics at the sentence level. The experiments on the benchmark data set and MOOC forum data set state that SMM outperforms baselines, its interaction mechanism is effective and it has an advantage in cross-domain duplicate question detection.

INDEX TERMS Community question answering, duplicate question detection, semantic matching, transfer learning.

I. INTRODUCTION

The CQA forum plays an important role in helping users find solutions to their questions and promoting online learning. Forums like Stack Overflow, Stack Exchange usually contain different domain information, while forums in MOOC[1] are usually managed by independent courses, and forums for different courses involve different professional knowledge. The CQA forum offers information retrieval to users to search before asking, but it still suffers from large numbers of duplicate questions, which have been answered before. The number of duplicate questions on Stack Overflow has been increasing over time [1]. Mismatch of duplicate questions may make users conceive that their questions have not been asked before and thus cause them to ask ''new'' questions, which are unlikely to be answered (because the original one has been answered). The continuous accumulation of duplicate questions may interfere with their information searching again, affecting user experience and satisfaction. Stack Overflow asks professional users to mark duplicate questions while online learning forums like the MOOC forum require teachers and teaching assistants to manage the posts, which inevitably wastes manpower and time. Automatic detection of duplicate questions is necessary and critical for CQA.

The essential task of duplicate question detection is to parse the semantics and logic of questions to identify whether two given questions are duplicate, corresponding to the task of semantic matching. Whereas, in the CQA forum, there are all kinds of questions, some of which correspond to the same answers, but with different ways of expression, some of which are with similar syntax and sentence patterns but have semantic gaps, bringing difficulties to the task. Fig. 1 shows the examples.

The associate editor coordinating the review of this manuscript and approving it for publication was Choon Ki Ahn.

[1]In the MOOC forum, students are encouraged to communicate and interact with each other. The forum plays an important role in asking for help and communication. Based on this setting, MOOC forum herein is classified as CQA forum.

why the vector theta is perpendicular to the boundary?

why the vector theta is at 90 degrees to the boundary?

(a) Duplicate questions with similar expressions.

[Bug] Not able to see quiz options.

Seem like Quiz Question not properly displayed?

(b) Duplicate questions with different expressions.

FIGURE 1. Duplicate questions in CQA forums.

In Fig. 1 (a), two sentences are semantically equivalent but they use different words to express the meaning of ''perpendicular''. Identifying the paraphrase of these two questions needs to overcome the lexical gap (semantic gap) [2], [3]. Feature-based approaches designed various text features, such as topic similarity, lexical similarity, syntactic features, etc. to learn semantic relationships between sentences [4]–[7], while model-based approaches tend to learn end-to-end classifiers for semantic matching [8]–[13]. In model-based approaches, effective components were designed to capture information denoting sentence similarity. The model-based approaches include two types of models, the representation-based model and the interaction-based model [14]. But whether previous works are feature-based or model-based, they tend to focus the semantic gap between words, which may ignore semantics that should be learned from the sentence level [15]. Fig. 1 (b) shows an example of duplicate questions in which semantics cannot be parsing just based on lexical gaps. The questions are with different expressions and few overlap words. Different expressions and unimportant words like ''[bug]'', ''seem like'' interfere with semantic understanding. Capturing the interaction between words is not enough for intricate cases in CQA forums.

In addition to the difficulties above, identifying duplicate questions on CQA forums also requires overcoming the problem of domain specificity. As mentioned before, CQA forums involve different domain knowledge. Different domains in CQA forums may have different contexts for semantic parsing, and most domains lack sufficient data for training [16]. A classifier trained on one domain is hard to achieve the same performance to predict duplicates in another domain [17].

At present, a small number of studies [2], [17]–[19] discussed domain transferability of duplicate question detection, and most of them achieve transfer learning through INIT strategies [20]. The hCNN [18] is the first work to apply multi-task transfer learning to domain-specific semantic matching (Paraphrase Identification). Nevertheless, this research didn't discuss the application on CQA forums.

To solve the above problems, this paper proposed a Semantic Matching Model (SMM) with multi-task transfer learning for multi-domain forum duplicate question detection. Based on the interaction between words, this method further extracted the interaction between words and sentences, which considered overall weights of words at the sentence level, and then extracted contextual information, better abstracting semantics from the entire sentence. By implementing the models in the multi-task framework, the domain transferability of SMM and baselines were compared. The experimental results show that SMM can effectively extract the interactive relationships between sentences and it is well suited for duplicate question detection in small domains.

The rest of this paper is organized as follows: Part II introduces the related work on duplicate question detection and semantic matching; Part III introduces the multi-task framework and the SMM; Part IV reports experimental settings and analyzes the results; The Part V summarizes.

II. RELATED WORK

Stack Overflow mining challenges in 2013 [21] and 2015 [22] gave rise to DupPredictor [5] and Dupe [4]: DupPredictor developed a framework to identify duplicates. Similarities of post titles, descriptions, topics and tags were considered as features to measure the overall similarity scores of questions. Dupe used features including term overlaps, entity and entity type overlaps, WordNet [23] similarity to train a classifier.

Extracting important features that describe text similarity is a key step. Whereas, there are some defects in these conventional methods: Zhang *et al.* [6] observed that duplicate classifiers just based on word vectors and topic similarities would fail to identify certain duplicate pairs, and they mined association phrases that co-occur frequently and used the association score as one of the features to make up for the defect of conventional methods. Silva *et al.* [1] reproduced DupPredictor [5] and Dupe [4] to detect duplicate questions in Stack Overflow, and they discovered that the effectiveness of the models was not as good as the original work. Hoogeveen *et al.* [16] used metadata such as user reputations, user behavior to help improve the detection of misflagged duplicate questions, and their experimental results stated that metadata from CQA forums can be integrated with text features to achieve good results. However, in these methods, features need to be designed manually, and the roles of manual constructed features may change with different data sets and tasks. Difficulties also exist in reproduction [1].

Different from the above work, Addair [24] explored the performance of neural networks in detecting duplicate questions. The experimental results showed that Multi-Layer Perception (MLP), siamese CNN and LSTM perform better than traditional NLP methods. Zhang *et al.* [2] took advantage of BiLSTM in NLP tasks and integrated it with FrameNet [25] to improve the detection.

The design of NN models has evolved from presentation-based models to interaction-based models [14]. Typical presentation-based models such as DSSM [26], CDSSM [27] extracted representations of sentences and then compared the similarity of text representations to measure the degree of matching. Interaction-based models aimed to extract effective matching patterns through the text-to-text interaction. A line of work, such as ARC-II [28], MatchPyramid [11], Conv-KNRM [13], RI-Match [15] designed interaction structures to describe the degree of text matching. Another line

of work, such as DAM [29], DRCN [8], MIX [14] used the Attention mechanism to learn word alignments.

The work of [15] mentioned the defect of the word-to-word interaction, it used BiLSTM to encode information from sentence level to complement the word-to-word interaction. Chen *et al.* [14] proposed a multi-channel crossing model integrating Inverse Document Frequency (IDF), Part-of-Speech (PoS) tags and word positions to combine local and global matching in text. Zhang *et al.* [2] and Chen *et al.* [14] both observed the importance of matching crucial parts in the sentence and they introduced external knowledge to help improve semantic matching.

Nevertheless, previous works tended to focus on semantic gaps between words. There are few discussions on how to design effective interaction mechanisms to abstract semantics of the whole sentence without introducing external knowledge. They were mainly to put forward solutions of semantic matching but seldom discussed the domain adaption of semantic matching.

Lan and Xu [10] explored the transferability of different semantic matching models. They trained the models on a source domain and test their performance on another domain. According to their analysis, previous interaction-based models were easy to focus on a few different words in the sentences with more overlap words. The adversarial network [19] was also used to achieve domain adaptive detection, but it had no noticeable advantage when the source and target domains were not similar enough.

The work of [18] is the first work to propose multi-task transfer learning to multi-domain semantic matching. This work designed hCNN which combines BCNN [12] and Pyramid [11] to capture the text interaction. Their experimental results proved that the combination of CNN and Pyramid is efficient and effective, and it can be better adapted to multiple domains through multi-task transfer learning. On the basis of hCNN [18], this paper proposed SMM, and with the help of multi-task transfer learning, the model was well adapted to multi-domain duplicate question detection.

Compared to related work, the contributions of this work are as follows:

1) This paper designed a mechanism to capture word-to-sentence interactions based on word-to-word interactions. The experimental results proved the effectiveness of the interaction mechanism.
2) It is the first work to apply multi-task transfer learning to duplicate question detection in CQA forums (especially in the educational forum, the MOOC forum).
3) State-of-the-art semantic matching models were reproduced in the transfer learning framework and adapted to the task of duplicate question detection. The model proposed by this paper outperforms the baselines.

III. METHODS

A. FRAMEWORK

Multi-domain duplicate question detection is considered as a binary classification task in this paper. In the framework,

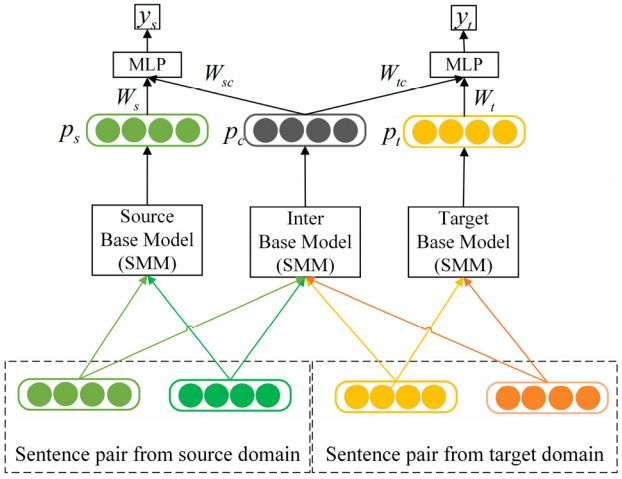


FIGURE 2. Multi-task transfer learning framework.

there is a base model identifying whether two questions are duplicate. The base model is responsible for extracting matching information of two given sentences and transforming such information through MLP with softmax to produce a label $y \in \{0, 1\}$, where y denotes whether the two sentences are matched. When a sentence pair is matched, the label $y = 1$, and otherwise, $y = 0$. To adapt the base model to multiple domains, the base model is integrated with the multi-task transfer learning framework proposed by [18]. As shown in Fig. 2, the framework contains source domain, inter-domain and target domain, all of which use the base model as the basic component.

Given sentence pairs Q_1 and Q_2, the base model is trained for the source, target and inter-domain, which will produce domain-specific output features p_s, p_c, p_t, respectively (as in Fig. 2). The source and the target domain share the information from the inter-domain to infer the domain-specific labels according to (1), where $W_d (d \in \{sc, s, st, t\})$ is the weight matrix of domain-specific MLP.

$$p(y \mid p_s, p_c, p_t) = \begin{cases} softmax\,(W_{sc}p_c + W_s p_s)\ if\ \ y\ from\ src \\ softmax\,(W_{tc}p_c + W_t p_t)\ if\ \ y\ from\ tgt \end{cases} \tag{1}$$

The framework transfers knowledge by a covariance matrix $\Omega \in \mathbb{R}^{4\times4}$ which constrains the relationships between weight matrices in W, where $W = [W_s; W_{sc}; W_{tc}; W_t]$, each column of which is domain specific W_d as mentioned before. Note that, each value in Ω is the similarity score of any two weight matrices in W. Therefore, by minimizing the trace of $W\Omega^{-1}W^T$ according to (2), entries in Ω will constrain the relationship between W_d. When the corresponding value is large, the relationship between W_d is strong, and otherwise, the relationship between them is relatively weak.

$$\mathcal{L}_1 = tr\left(W\Omega^{-1}W^T\right) \quad s.t. \Omega \geq 0, \quad tr\,(\Omega) = 1. \tag{2}$$

Cooperating with constrain domain relationships, the model updates parameters by minimizing the loss function $\mathcal{L}$,

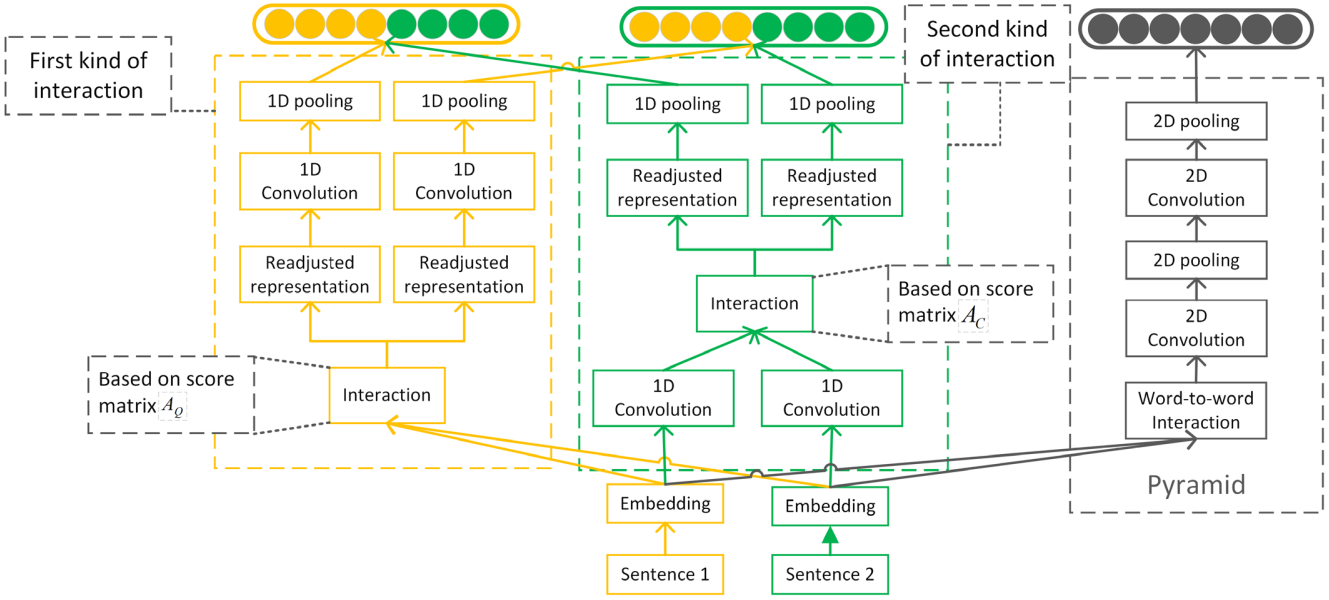


FIGURE 3. The structure of semantic matching model (the element-wise comparison and classification layer are not shown).

where n_d denotes the number of training samples of the source or the target domain, λ_1 is the loss weight of $\mathcal{L}_1$. $\mathcal{L}_2$ denotes L_2 regularization of all parameters.

$$\mathcal{L} = -\frac{1}{n_d}\sum_{i}^{n_d}\log p\left(y_i \mid Q_1^i, Q_2^i\right) + \frac{\lambda_1}{2}\mathcal{L}_1 + \mathcal{L}_2, d \in \{src, tgt\} \quad (3)$$

B. SEMANTIC MATCHING MAODEL

The Semantic Matching Model (SMM) is the base model of the framework. It is composed of four main components, which are the context representation layer, the interaction layer, the Pyramid architecture [11], [18] and the classification layer. The context representation layer takes two input sentences and encodes the sentence with the word embedding and 1D CNN. The interaction layer takes context representations from the Embedding layer and the 1D CNN layer to calculate two kinds of word-to-word interaction patterns and then further extract word-to-sentence interactions to readjust their contextual representations. The Pyramid is used to capture the element-wise word-to-word interaction. Finally, the outputs of the interaction layer are further compared and integrated with the output of the Pyramid to produce richer matching patterns for classification.

1) CONTEXT REPRESENTATION

As shown in Fig. 3, the Embedding layer produces two representations, $Q_1 = \left(q_1^1, q_2^1, \ldots, q_n^1\right)$ and $Q_2 = \left(q_1^2, q_2^2, \ldots, q_m^2\right)$, where q_i^j denotes a d-dimension embedding vector of a word, i denotes the position of a word in the sentence and j denotes which of the two sentences the word comes from. 1D CNN is used to produce the higher-level context representation C. After the convolutional kernel slides over the sentence embedding, 1D CNN convolutes the sentence as the following representation. While CNN has k kernels, c_i^j will be a k dimension vector.

$$C_1 = CNN\left(Q_1\right) = \left(c_1^1, c_2^1, \ldots, c_{n'}^1\right)$$
$$C_2 = CNN\left(Q_2\right) = \left(c_1^2, c_2^2, \ldots, c_{m'}^2\right) \quad (4)$$

2) INTERACTION LAYER

Given the output $X_1 \in \{Q_1, C_1$ and $X_2 \in \{Q_2, C_2$ from the context representation layer, the interaction layer first calculates word-to-word similarities as in (5) to produce a score matrix $A_Q \in R^{n \times m}$ for the Embedding layer, and $A_C \in R^{n' \times m'}$ for the CNN layer.

$$A_X = X_1^T X_2 \quad (5)$$

Herein, each element in the score matrix uses dot product of corresponding word vectors to measure the similarity between words.

To obtain more robust interactive information, the model further calculates the word-to-sentence interaction based on A_Q and A_C. Note that in the score matrix, the elements in the row i are unnormalized similarity scores of all x_j^2 to the x_i^1, and the elements in the column j are unnormalized similarity scores of all x_i^1 to the x_j^2. Therefore, by summing each score in the row and normalizing the score with column-wise softmax as in (6), the overall weight of representation X_2 to each word in X_1 can be obtained. In the same way, by summing each score in the column and normalizing the score with row-wise

softmax as in (7), the overall weight of representation X_1 to each word in X_2 can be obtained.

$$\alpha_{x_i} = \frac{exp(\sum_j A_{x_{ij}})}{\sum_i exp(\sum_j A_{x_{ij}})} \tag{6}$$

$$\beta_{x_j} = \frac{exp(\sum_i A_{x_{ij}})}{\sum_j exp(\sum_i A_{x_{ij}})} \tag{7}$$

The weights α and β of each element for two given sentences can be calculated as above. After obtaining α and β, we readjust the original representation X according to (8), where X_1' and X_2' denote the adjusted representations of X_1 and X_2, respectively.

$$\begin{aligned} X_1' &= (\alpha_{x_1} x_1^1, \alpha_{x_2} x_2^1, \alpha_{x_3} x_3^1, \ldots\ldots) \\ X_2' &= (\beta_{x_1} x_1^2, \beta_{x_2} x_2^2, \beta_{x_3} x_3^2, \ldots\ldots) \end{aligned} \tag{8}$$

Note that representations of Embedding and CNN layers are both considered as X, thus, the readjusted embedding representations Q_1', Q_2' can be calculated, and the readjusted CNN representations C_1', C_2' can also be calculated. In other words, we get two kinds of word-to-sentence interaction information, one from the Embedding, as the left part in Fig. 3, and another from the CNN layer, as the middle part in Fig. 3.

After the above steps, the information that is relatively important to each other can be extracted according to their mutual word-to-sentence scores.

To further extract contextual information of the readjusted representation Q_1', Q_2', 1D CNN as mentioned above is used to convolute Q_1' and Q_2' and produce higher-level contextual representation C_1'', C_2''. Furthermore, 1D all-pooling is applied to C_1', C_2', C_1'' and C_2''. Finally, the corresponding outputs of pooling are concatenated for two given sentences, as in (9).

$$\begin{aligned} z_1 &= pooling\left(C_1'\right) \oplus pooling(C_1'') \\ z_2 &= pooling\left(C_2'\right) \oplus pooling(C_2'') \end{aligned} \tag{9}$$

Herein, 1D all-pooling is the max-pooling as in [18], and $\oplus$ denotes the operation of concatenating between outputs. Furthermore, the model performs element-wise comparison [30], [31] of z_1 and z_2 to obtain the final representation z_v of the interaction layer.

3) PYRAMID

The Pyramid architecture [11] is used to extract matching patterns from the word-to-word interaction produced in III-B-2) since the combination of Pyramid architecture and CNN has been proved to be effective and efficient [18]. As shown in Fig. 3 (the rightmost part), Pyramid herein is composed of two 2D CNN and two 2-D max-pooling. Because the rows and columns of the score matrix correspond to words from two sentences, 2D CNN can extract such interactive information from two dimensions, capturing the relationship between multiple words and phrases in two sentences. The output of such a relationship is represented by a 2D matrix. Finally, we squeeze the 2D matrix into a 1D vector as the final output of the Pyramid.

4) CLASSIFICATION LAYER

The output of the interaction layer z_v and the output of the Pyramid z_p are concatenated to produce final output p for classification. Note that the model is trained for three domains, each of which would produce a domain-specific output. Thus, we would get three output features p_s, p_c, p_t for MLP with softmax to produce two domain-specific labels as mentioned in III-A.

IV. EXPERIMENTS

This section introduces the data sets of the experiment, implemental details of SMM and experimental settings of model comparisons. The performance of SMM and baseline algorithms is reported.

A. DATA SETS

In order to evaluate the performance of the models in detecting domain duplicate questions, we took the released benchmark data set CQADupStack [32] and our private MOOC data set as the target domain, and the benchmark large-scale data set Quora[2] as the source domain.

CQADupStack contains twelve subforums from Stack Exchange. We randomly selected five domain-specific forums, which are Games, Mathematics, Android, Statistics and Physics for experiments. All duplicate questions in each domain were extracted by query_cqadupstack,[3] and each duplicate question pair is combined with non-duplicate question randomly selected from the non-duplicate set to form the final data set with a positive to negative ratio of 1:3, simulating a scenario of skew distribution of duplicates and non-duplicates in the forums. The amount of data in each domain is less than 10,000. The MOOC data set comes from a Machine Learning (ML) course in Coursera.[4] We asked domain professors and students to tag question pairs on potentially similar question set, forming a MOOC data set with about 1,000 questions pairs and a positive to negative ratio of about 1:1. Detailed information of data sets is shown in Table 1.

All data in the target domain were randomly shuffled and split into a train set and a test set with a ratio of 8:2. Accuracy (ACC) and F1 are used as the main evaluation metrics to compare model performance in detecting duplicate questions.

B. MODEL IMPLEMENTATION

The word embedding was initialized with 300-dimensional GloVe [33] vectors which are pretrained in the 840B Common Crawl corpus. The Embedding was set to be trainable. The threshold for the length of the input sentence was set to 100. The kernel size and the number of kernels for CNN in context representation layers were set to 4 and 150 respectively, with strides set to 1. For two CNN layers in Pyramid, kernel sizes

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

[3] https://github.com/D1Doris/CQADupStack/blob/master/query_cqadupstack.py

[4] https://www.coursera.org/

TABLE 1. Details of data sets. the "details" column describes the amount of data in the format of "total/positive+".

	Domain /Discipline	Details (Total/Positive+)	Model Inputs
StackExchange	Game	9,116/2,279+	Target
	Math	5,488/1,372+	
	Android	6,852/1,713+	
	Stats	3,664/916+	
	Physics	7,872/1,968+	
MOOC	ML	1,016/553+	
Quora[1]	No distinction	404,303/ 149,265+	Source

were set to 4×4 and 2×2, and strides were set to 1 and 3. The strides of the 1D max-pooling layer and two 2D max-pooling layers were set to 1, 4 and 2 respectively. λ_1 was set to 0.0004. Additionally, the batch size and learning rate were set to 64 and 0.08. Training of transfer learning framework followed the algorithm proposed by [18]. Each model was tested 5 times on the test set in each domain and the averages of all tested results were recorded as the final results.

C. EXPERIMENTAL SETTINGS AND RESULTS

SMM was compared with the state-of-the-art CNN-based and LSTM-based models. By setting up three groups of experiments, the performance in the compared methods, the effects of the interaction mechanism and the transferability of SMM were examined. For fairness, all compared methods were adapted to the multi-task transfer learning framework.

1) BASELINES

(1) **CNN** is the context representation layer in SMM.
(2) **Pyramid** [11] is the basic component in SMM.
(3) **hCNN** is the first hybrid CNN model proposed by [18].
(4) **SMM-att**: Replace the interaction layer in SMM with the general attention mechanism [34] to compare the interaction mechanism proposed with the attention mechanism.
(5) **Siamese LSTM** is proposed by [35] for learning semantic similarity. In the experiments, MaLSTM is used for the abbreviation of **Siamese LSTM**.

Conv-KNRM is the state-of-the-art CNN-based n-gram matching model proposed by [13]. In order to compare the matching strategy proposed by [13] with the interaction mechanism, the n-gram cross-matching (abbreviated as NGM) structure was extracted from Conv-KNRM and served as a component in the following methods. The lengths of n-grams were set to 1, 2, 3 according to [13].

(6) **NGM1**: Use n-gram matching in [13] to produce score matrix A (as mentioned in III-B-2)) and use Pyramid to capture higher-level signals of n-gram matching.
(7) **NGM2**: Use n-grams matching in [13] to produce score matrix A (as mentioned in III-B-2)) and use the interaction mechanism mentioned in III-B-2) to extract interaction between n-grams and sentences. The other structures in the model remained unchanged.

The experimental results are shown in Table 2. Table 3 summarizes the results of the comparison between SMM and baseline algorithms.

By comparing the results of CNN, Pyramid, hCNN and SMM, it can be concluded that integrating CNN with Pyramid can bring significant benefit to the detection. Though the performance of the Pyramid is not as good as that of SMM, its effectiveness can't be ignored.

It can be observed that the performance of SMM exceeds that of MaLSTM in all six domains, and it outperforms hCNN in Games, Mathematics, Statistics, Physics and MOOC, four of which are improved significantly. Although no improvement of ACC is observed on Android, its F1 value is 0.7% higher than that of hCNN. SMM outperforms hCNN in the F1 with a maximum improvement of 9.5%. MaLSTM has no interaction mechanism, and hCNN lacks the word-to-sentence interaction proposed in this paper. The experimental results indicate that the word-to-sentence interaction is important to semantic matching.

Compared to SMM-att, SMM is different in the way it calculates the interactive information of sentences. From the results in Table 2, SMM-att does not perform as well as SMM in most domains except MOOC, which means that capturing interactive information through Attention is not necessarily effective in cross-domain duplicate question detection. Attention considers word alignments, which may let the model focus on word-to-word interactions and neglect relatively the attention from the sentence level, while the word-to-sentence interaction summarizes relatively attention from the sentence level and uses overall weights to readjust the representation of the sentences. The experimental results show that the effects of these two mechanisms are different, and the word-to-sentence interaction mechanism is more effective than the Attention mechanism.

The difference between the results of NGM1 and NGM2 further supports the above basic conclusion: The strategy of the n-gram matching cannot capture matching patterns better than the word-to-sentence interaction does, since integrating n-grams with the interaction mechanism (NGM2) outperforms NGM1. Compared to SMM, NGM2 has improvements in Game, Math and Android, but not in Stats, Physics and ML. The performance of NGM2 and SMM is comparable. It is known that when n is 1, the matching is word level, and when n is greater than 1, the matching is phrase level. Thus, the results above indicate that there seems to be little benefit to use interactive information of both word and phrase.

In conclusion, the comparisons between SMM and baselines demonstrate the effectiveness of SMM and its word-to-sentence interaction mechanism.

2) TWO KINDS OF INTERACTIONS

The interaction layer takes the representations from Embedding and CNN layers to produce two kinds of interactions. In the previous experiment, the interaction mechanism was

TABLE 2. Comparison between SMM and baselines. ACC and F1 in each domain and model are shown in the format of "ACC/F1", and the appearance of the symbol £ indicates there were significant differences between the performance of SMM and the performance of baselines.

	StackExchange					MOOC
	Game	Math	Android	Stats	Physics	ML
CNN	0.790£/0.540£	0.797£/0.465£	0.862£/0.682£	0.819£/0.493£	0.862£/0.671£	0.805£/0.795
Pyramid	0.901/0.796	0.804£/0.570£	0.893/0.773	0.853/0.645	0.900£/0.775£	0.795/0.786
hCNN	0.887/0.758	0.810/0.539£	0.907/0.791	0.837£/0.575£	0.905/0.775£	0.806£/0.797£
SMM-att	0.868£/0.719£	0.792/0.476£	0.895/0.762£	0.835£/0.597£	0.879£/0.712£	0.823/0.808
MaLSTM	0.775£/0.506£	0.775£/0.506£	0.852£/0.673£	0.791£/0.523£	0.860£/0.690£	0.771£/0.484£
NGM1	0.876£/0.741£	0.805£/0.517£	0.885£/0.738£	0.855/0.636	0.886£/0.746£	0.825/0.821
NGM2	**0.899/0.788**	**0.825/0.606**	**0.909/0.802**	0.855/0.651	0.909/0.791	0.821/0.813
SMM	**0.897/0.781**	**0.821/0.596**	0.906/0.798	**0.863/0.670**	**0.912/0.799**	**0.825/0.818**

TABLE 3. Summary of comparison between SMM and baselines. When SMM has improvements in domains, the domain color is red, and the more the improvement, the darker the color. When SMM is worse than baselines, the color is green. When there is no improvement, the color is the middle color between red and green.

	Domains						Improvement of SMM in six domains Format: Average (min ~ max)		Model features of baselines compared to SMM
	Game	Math	Android	Stats	Physics	MOOC	ACC	F1	
CNN							4.8% (2%~10%)	13.6% (2.3%~24.1%)	Without interaction and Pyramid
Pyramid							1.3% (-0.4%~3%)	1.95% (-1.5%~3.2%)	Without CNN context layer and interaction
hCNN							1.2% (-0.1%%~2.6%)	3.8% (0.7%~9.5%)	Without interaction mentioned in III-B-2)
SMM-att							2.2% (0.2%~3.3%)	6.5% (-0.3%~12%)	Replacing interaction with Attention
MaLSTM							6.6% (4.6%~12.2%)	18% (9%~27.5%)	Base model is different
NGM1							1.5% (0.8%~2.6%)	4.4% (3.4%~7.9%)	With n-gram matching
NGM2							0.1% (-0.4%~0.8%)	0.18% (-0.9%~1.9%)	With interaction mentioned in III-B-2)

TABLE 4. Comparison of EMB-I and CNN-I. ACC and F1 in each domain and model are shown in the format of "ACC/F1".

		EMB-I	CNN-I
Stack Exchange	Game	0.881/0.750	0.899/0.788
	Math	0.806/0.567	0.823/0.605
	Android	0.897/0.781	0.906/0.796
	Stats	0.839/0.625	0.870/0.691
	Physics	0.890/0.747	0.909/0.791
MOOC	ML	0.787/0.778	0.825/0.814

proved to be effective, but what is the role of these two kinds of interactions? We removed one of them to produce two models, EMB-I and CNN-I, and examined their effects on the task. Note that EMB-I and CNN-I are also adapted to the transfer learning framework.

(1) **EMB-I** is the word-to-sentence interaction based on the Embedding layer.
(2) **CNN-I** is the word-to-sentence interaction based on the CNN layer.

From the results in Table 2 and 4, it can be observed that the performance of EMB-I is worse than SMM, and the performance of CNN-I is comparable to SMM. It seems that the interaction of the Embedding layer brings little benefit to SMM. The result of using the CNN context to produce the word-to-sentence interaction is better than that of using the Embedding context directly. Whereas, the results of such a combination are not worse than that of CNN-I, which indicates that the combination of two kinds of interactions has the effect of $1 + 1 \neq 2$.

3) TRANSFERABILITY OF SMM

In this section, we compare the performance of SMM with and without multi-task transfer learning to examine its transferability. Since SMM is proposed based on hCNN, we just make a comparison between SMM and hCNN to see how much SMM can improve the transferability of its basic model. The results are shown in Table 5. *Origin* denotes the model that was trained and tested in the target domain, and *Trans* denotes the model that was trained and tested in the multi-task transfer learning framework. By comparing the results between *Origin* and *Trans*, the transferability of the model can be learned: Through the framework, hCNN achieves 0.09% to 2.2% improvements of ACC and 2.2% to 5% improvements of F1 in all domains except MOOC. SMM improves ACC in all domains by 1.4% to 3.2% and improves F1 by 2.8% to 7.6% compared to its Origin model. Its transferability is stronger than that of hCNN.

4) VISUAL ANALYSIS

The covariance matrix Ω in the five domains on Stack Exchange and score matrices of question pairs were extracted and visualized to better investigate the performance of SMM.

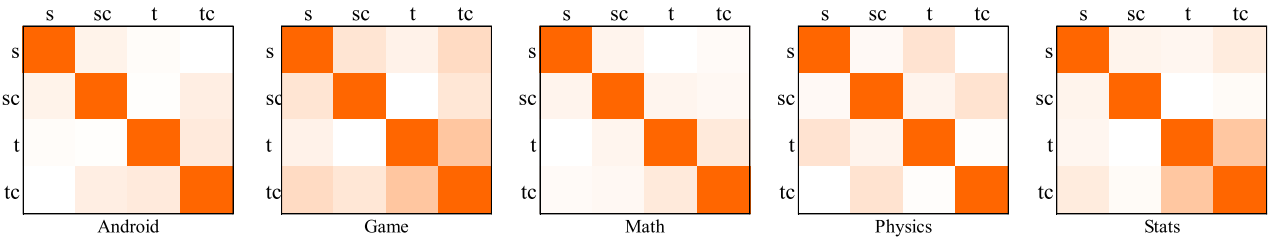
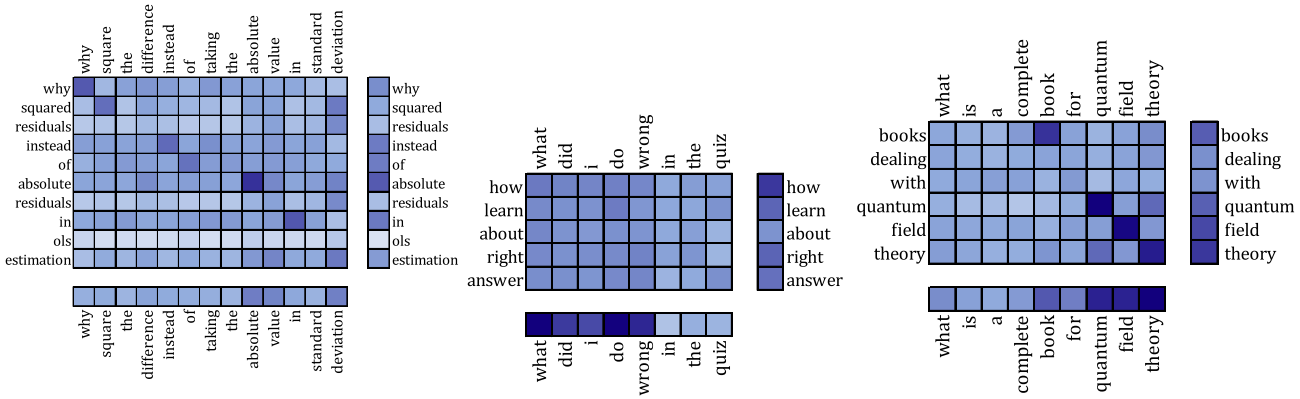
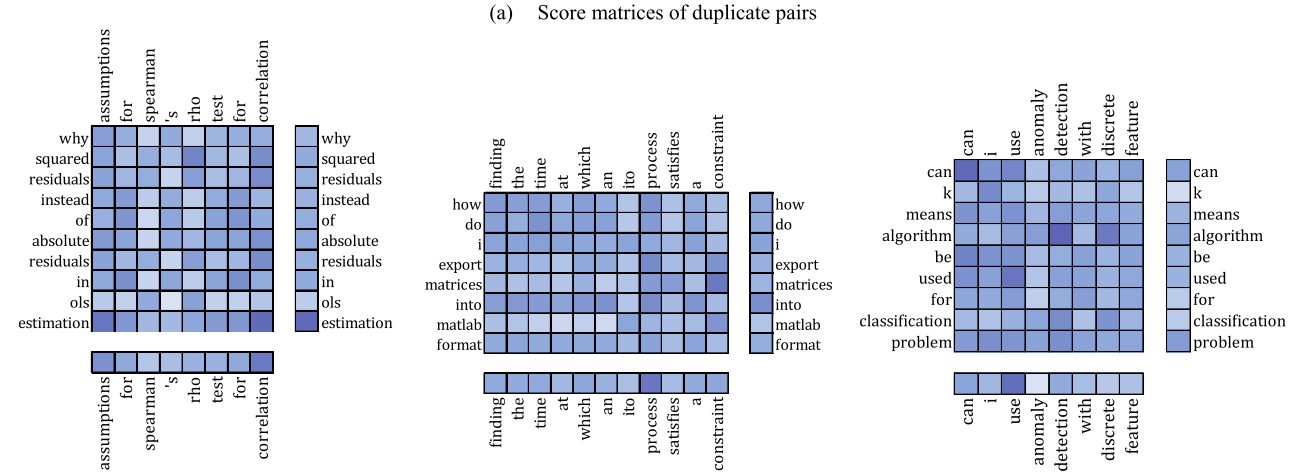


FIGURE 4. Heat maps of covariance matrix Ω in the transfer learning framework (in III-A). The darker the color, the larger the entry value. Note that Ω reflect relationships between W_s; W_{sc}; W_{tc}; W_t, which denote the weights of the source, shared-to-source, shared-to-target and target domain.



(a) Score matrices of duplicate pairs



(b) Score matrices of non-duplicate pairs

FIGURE 5. Heat maps of score matrices A_X in the interaction layer (in III-B). The darker the color, the larger the entry value.

TABLE 5. The transferability of SMM and hCNN. Origin denotes the model that was trained and tested in the target domain, and Trans denotes the model that was trained and tested in the multi-task transfer learning framework.

		StackExchange					MOOC
		Game	Math	Android	Stats	Physics	ML
hCNN	Origin	0.870/0.724	0.801/0.511	0.883/0.744	0.824/0.525	0.883/0.732	0.814/0.809
	Trans	0.887/0.758	0.810/0.539	0.907/0.791	0.837/0.575	0.905/0.775	0.806/0.797
SMM	Origin	0.872/0.730	0.807/0.568	0.879/0.739	0.831/0.594	0.887/0.745	0.798/0.785
	Trans	0.897/0.781	0.821/0.596	0.906/0.798	0.863/0.670	0.912/0.799	0.825/0.818

In Fig. 4, heat maps of Ω are plotted, each of which is the average result of multiple matrices that randomly extracted from its domain. In Fig. 5, word-to-word score matrices of duplicate pairs and non-duplicate pairs are also plotted by heat maps. On the right and bottom of each subgraph, word-to-sentence scores are shown, which helps to better understand the word-to-sentence interaction.

In Fig. 4, the relationship between W_{sc} and W_t, W_{tc} and W_s are both weak. The observation is in line with empirical cognition that the feature space shared to the source

domain should be different from the feature space of the target domain, and this relationship also exists in the feature space of the source domain and the shared-to-target domain. It noteworthy that W_{sc} and W_{tc} are both from the inter-domain, they should be correlated to each other, and the relationship does occur in most domains. The relationship between W_{sc} and W_s, W_{tc} and W_t are strong. This result is not consistent with that described by [18], which means that the relationship is not immutable and can be affected by the data set.

Heat maps in Fig. 5 (a) and (b) show a clear distinction between similar words, co-occurrence words and dissimilar words. Colors in Fig 5. (a) are darker than that in Fig. 5 (b), which means that word similarities in duplicate pairs are stronger than that in non-duplicate pairs. The right and bottom of each subgraph illustrate the word-to-sentence interaction can identify keywords in duplicate pairs and ignore potential similar words in non-duplicate pairs.

V. CONCLUSION AND FUTURE WORK

Identifying semantic equivalence of sentences is essential to duplicate question detection. Although semantic similarity can be learned by measuring the similarity between words, intricate relationships between semantics, the ways of expression and syntax make the task more challenging. This paper proposes SMM, which abstracts semantics based on the interaction mechanism. It takes interactive information from the Embedding and the CNN layer respectively to capture abundant matching information for semantic parsing. By cooperating with the multi-task transfer learning framework, SMM can be well applied to domain-specific forums. By reweighting word-to-word scores at the sentence level and using reweighted scores to readjust the original representation, the interaction mechanism further extracts keywords for duplicate pairs and ignores potential similar words for non-duplicate pairs. The experiments on the benchmark data set and MOOC forum data show that the combination of the word-to-sentence interaction, CNN and Pyramid is effective.

Although the interaction mechanism plays an important role in semantic matching, the addition of the interaction mechanism would reduce the efficiency of the hybrid model, which might require a tradeoff between efficiency and other performance indicators in practice.

In future work, we consider doing more experiments on model training time and inference time to examine the efficiency of the model. Exploring what kind of knowledge is transferred from the source domain to the target domain and giving an interpretable analysis are also the focuses of our future work.

REFERENCES

[1] R. F. Silva, K. Paixão, and M. de Almeida Maia, "Duplicate question detection in stack overflow: A reproducibility study," in *Proc. IEEE 25th Int. Conf. Softw. Anal., Evol. Reeng. (SANER)*, Mar. 2018, pp. 572–581.

[2] X. Zhang, X. Sun, and H. Wang, "Duplicate question identification by integrating framenet with neural networks," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, pp. 6061–6068.

[3] D. Bernhard and I. Gurevych, "Answering learners' questions by retrieving question paraphrases from social Q&A sites," in *Proc. 3rd Workshop Innov. NLP Building Educ. Appl.*, Jun. 2008, pp. 44–52.

[4] M. Ahasanuzzaman, M. Asaduzzaman, C. K. Roy, and K. A. Schneider, "Mining duplicate questions in stack overflow," in *Proc. 13th Int. Workshop Mining Softw. Repositories (MSR)*. New York, NY, USA: ACM, 2016, pp. 402–412.

[5] Y. Zhang, D. Lo, X. Xia, and J.-L. Sun, "Multi-factor duplicate question detection in stack overflow," *J. Comput. Sci. Technol.*, vol. 30, no. 5, pp. 981–997, Sep. 2015.

[6] W. E. Zhang, Q. Z. Sheng, J. H. Lau, and E. Abebe, "Detecting duplicate posts in programming QA communities via latent semantics and association rules," in *Proc. 26th Int. Conf. World Wide Web (WWW)*, 2017, pp. 1221–1229.

[7] C. Liang, P. K. Paritosh, V. Rajendran, and K. D. Forbus, "Learning paraphrase identification with structural alignment," in *Proc. IJCAI*, 2016, pp. 2859–2865.

[8] S. Kim, I. Kang, and N. Kwak, "Semantic sentence matching with densely-connected recurrent and co-attentive information," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, Jul. 2019, pp. 6586–6593.

[9] Y. Gong, H. Luo, and J. Zhang, "Natural language inference over interaction space," 2017, *arXiv:1709.04348*. [Online]. Available: http://arxiv.org/abs/1709.04348

[10] W. Lan and W. Xu, "Neural network models for paraphrase identification, semantic textual similarity, natural language inference, and question answering," in *Proc. 27th Int. Conf. Comput. Linguistics*, 2018, pp. 3890–3902.

[11] L. Pang, Y. Lan, J. Guo, J. Xu, S. Wan, and X. Cheng, "Text matching as image recognition," in *Proc. 13th AAAI Conf. Artif. Intell.*, 2016, pp. 3890–3902.

[12] W. Yin, H. Schütze, B. Xiang, and B. Zhou, "ABCNN: Attention-based convolutional neural network for modeling sentence pairs," *Trans. Assoc. Comput. Linguistics*, vol. 4, pp. 259–272, Dec. 2016.

[13] Z. Dai, C. Xiong, J. Callan, and Z. Liu, "Convolutional neural networks for soft-matching n-grams in ad-hoc search," in *Proc. 11th ACM Int. Conf. Web Search Data Mining*, vol. 2018, pp. 126–134.

[14] H. Chen, F. X. Han, D. Niu, D. Liu, K. Lai, C. Wu, and Y. Xu, "MIX: Multi-channel information crossing for text matching," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*. New York, NY, USA: ACM, Jul. 2018, pp. 110–119.

[15] L. Chen, Y. Lan, L. Pang, J. Guo, J. Xu, and X. Cheng, "RI-match: Integrating both representations and interactions for deep semantic matching," in *Proc. Asia Inf. Retr. Symp.* Cham, Switzerland: Springer, 2018, pp. 90–102.

[16] D. Hoogeveen, A. Bennett, Y. Li, K. M. Verspoor, and T. Baldwin, "Detecting misflagged duplicate questions in community question-answering archives," in *Proc. 12th Int. AAAI Conf. Web Social Media*, 2018, pp. 112–120.

[17] M. Shazan Mohomed Jabbar, L. Kumar, H. Samuel, M.-Y. Kim, S. Prabhakar, R. Goebel, and O. Zaïane, "On generality and knowledge transferability in cross-domain duplicate question detection for heterogeneous community question answering," 2018, *arXiv:1811.06596*. [Online]. Available: http://arxiv.org/abs/1811.06596

[18] J. Yu, M. Qiu, J. Jiang, J. Huang, S. Song, W. Chu, and H. Chen, "Modelling domain relationships for transfer learning on retrieval-based question answering systems in E-commerce," in *Proc. 11th ACM Int. Conf. Web Search Data Mining (WSDM)*. New York, NY, USA: ACM, 2018, pp. 682–690.

[19] D. J Shah, T. Lei, A. Moschitti, S. Romeo, and P. Nakov, "Adversarial domain adaptation for duplicate question detection," 2018, *arXiv:1809.02255*. [Online]. Available: http://arxiv.org/abs/1809.02255

[20] L. Mou, Z. Meng, R. Yan, G. Li, Y. Xu, L. Zhang, and Z. Jin, "How transferable are neural networks in NLP applications?" 2016, *arXiv:1603.06111*. [Online]. Available: http://arxiv.org/abs/1603.06111

[21] A. Bacchelli, "Mining challenge 2013: Stack overflow," in *Proc. 10th Work. Conf. Mining Softw. Repositories (MSR)*, 2013.

[22] A. T. T. Ying, "Mining challenge 2015: Comparing and combining different information sources on the Stack Overflow data set," in *Proc. 12th Work. Conf. Mining Softw. Repositories (MSR)*, 2015.

[23] G. A. Miller, "WordNet: A lexical database for English," *Commun. ACM*, vol. 38, no. 11, pp. 39–41, 1995.

[24] T. Addair, "Duplicate question pair detection with deep learning," Stanford Univ., Stanford, CA, USA, Tech. Rep. cs224n, 2017. [Online]. Available: https://web.stanford.edu/class/archive/cs/cs224n/cs224n.1174/reports/2748045.pdf

[25] C. F. Baker, C. J. Fillmore, and J. B. Lowe, "The berkeley FrameNet project," in *Proc. 17th Int. Conf. Comput. Linguistics*, vol. 1. Stroudsburg, PA, USA: Association for Computational Linguistics, 1998, pp. 86–90.

[26] P.-S. Huang, X. He, J. Gao, L. Deng, A. Acero, and L. Heck, "Learning deep structured semantic models for Web search using clickthrough data," in *Proc. 22nd ACM Int. Conf. Conf. Inf. Knowl. Manage. (CIKM)*, 2013, pp. 2333–2338.
[27] Y. Shen, X. He, J. Gao, L. Deng, and G. Mesnil, "A latent semantic model with convolutional-pooling structure for information retrieval," in *Proc. 23rd ACM Int. Conf. Conf. Inf. Knowl. Manage. (CIKM)*, 2014, pp. 101–110.
[28] B. Hu, Z. Lu, H. Li, and Q. Chen, "Convolutional neural network architectures for matching natural language sentences," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2042–2050.
[29] A. P. Parikh, O. Täckström, D. Das, and J. Uszkoreit, "A decomposable attention model for natural language inference," 2016, *arXiv:1606.01933*. [Online]. Available: http://arxiv.org/abs/1606.01933
[30] L. Mou, R. Men, G. Li, Y. Xu, L. Zhang, R. Yan, and Z. Jin, "Natural language inference by tree-based convolution and heuristic matching," 2015, *arXiv:1512.08422*. [Online]. Available: http://arxiv.org/abs/1512.08422
[31] S. Wang and J. Jiang, "A compare-aggregate model for matching text sequences," in *Proc. ICLR*, 2017, pp. 1–11.
[32] D. Hoogeveen, K. M. Verspoor, and T. Baldwin, "CQADupStack: A benchmark data set for community question-answering research," in *Proc. 20th Australas. Document Comput. Symp. (ADCS)*, 2015, pp. 1–8.
[33] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2014, pp. 1532–1543.
[34] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," 2014, *arXiv:1409.0473*. [Online]. Available: http://arxiv.org/abs/1409.0473
[35] J. Mueller and A. Thyagarajan, "Siamese recurrent architectures for learning sentence similarity," in *Proc. 13th AAAI Conf. Artif. Intell.*, 2016, pp. 2786–2792.

ZHUOJIA XU was born in Chaozhou, Guangdong, China, in 1994. She received the B.S. degree in computer science and engineering from the South China University of Technology, Guangzhou, Guangdong, in 2017, where she is currently pursuing the master's degree with the School of Computer Science and Engineering.

Since 2017, she has been engaged in natural language processing and educational data mining with the Communication and Computer Network Laboratory of Guangdong. Her two EI-index articles were published during her study period. Her research interests include student behavior mining in online learning and blended learning, forum information recommendation, and forum information retrieval.

Ms. Xu won the Third Place in the Government Big Data Governance Competition held by China Electronics Technology Group Corporation, in 2018.

HUA YUAN was born in Sichuan, China, in 1969. She received the M.S. and Ph.D. degrees from Sichuan University, Chengdu, Sichuan, in 1996 and 2001, respectively.

She has been an Associate Professor with the School of Computer Science and Engineering, South China University of Technology, since 2001. She has been engaged in IPv6/v4 video transmission, network image processing and retrieval, spam image filtering, IPv6 multicast, multimedia information processing and retrieval, and big data processing technology research. As a major researcher, she has participated in more than ten projects, such as the National Natural Science Foundation of China, the Natural Science Foundation of Guangdong Province, and the CNGI of the National Development and Reform Commission. She has published more than 50 professional academic articles. Her research interests include image processing, video communications, big data processing, next generation network architecture, online education, and blended learning.

Dr. Yuan won the First Prize for Excellent Undergraduate Teaching, the Nanguang Award for Excellent Undergraduate Teaching, and the Third Prize for Excellent Teaching and Research Papers from the South China University of Technology, in June 2015, June 2016, and December 2017.