One vs. Many QA Matching with both Word-level and Sentence-level Attention Network

Lu Wang[1, 2], **Shoushan Li**[1, 2, *], **Changlong Sun**[3], **Xiaozhong Liu**[3], **Luo Si**[3], **Min Zhang**[1,2], **Guodong Zhou**[1]

[1]NLP Lab, School of Computer Science and Technology, Soochow University, China
[2]Collaborative Innovation Center of Novel Software Technology and Industrialization
[3]Alibaba Group, China
[1]wanglu_1994@126.com, {lishoushan, minzhang, gdzhou}@suda.edu.cn
[3]{changlong.scl, xiaozhong.lxz, luo.si}@alibaba-inc.com

Abstract

Question-Answer (QA) matching is a fundamental task in the Natural Language Processing community. In this paper, we first build a novel QA matching corpus with informal text which is collected from a product reviewing website. Then, we propose a novel QA matching approach, namely One vs. Many Matching, which aims to address the novel scenario where one question sentence often has an answer with multiple sentences. Furthermore, we improve our matching approach by employing both word-level and sentence-level attentions for solving the noisy problem in the informal text. Empirical studies demonstrate the effectiveness of the proposed approach to question-answer matching.

1 Introduction

Question answer (QA) matching is a task to determine whether an answer is answering a given question. For instance, in Figure 1, the question "*Where is dear john filmed at?*" in **E1** has two candidate answers "*The movie was filmed in 2009 in Charleston.*" and "*The file was released on May 25, 2010 on DVD.*" The first answer is determined with the "*Matching*" category since it answers the question while the second answer is "*Non-matching*" since it could not answer the question. The past five years have witnessed a huge exploding interest in the research on QA matching, due to its widely applications, such as question answering (Yang et al., 2015; Tan et al., 2016; Wang et al., 2017) and reading comprehension (Trischler et al., 2016; Dhingra et al., 2017).

However, all existing QA matching studies only focus on formal text. In real applications, there exists many scenarios where the QA text is informal. For instance, **E2** is a question-answer pair ext-

E1: Two QA pairs in formal text	
Q_1: *Where is dear john filmed at?* Label: *Matching*	A_1: *The movie was filmed in 2009 in Charleston.*
Q_2:*Where is dear john filmed at?* Label: *Non-matching*	A_2: *The film was released on May 25, 2010 on DVD.*
E2: Three QA pairs in informal text	
Q: *Will the response time slow after updating os? What about the battery? What about the screen?*	A: *The response time is not a spark of slow, it has 4G RAM. But you must use it carefully, my phone's secreen has broke down.*
Q_1: *Will the response time slow after updating os?* Label: *Matching*	A_1: *The response ...... broke down.*
Q_2: *What about the battery?* Label: *Non-matching*	A_2: *The response ...... broke down.*
Q3: *What about the screen?* Label: *Matching*	A_3: *The response ...... broke down.*

Figure 1: Some QA examples with their matching labels

*Corresponding author.

racted from the product reviewing platform "*Asking All*" in *Taobao*[1]. The QA text is informal where the question "*Will the response time slow after updating os? What about the battery? What about the screen?*" has more than one question and the answer contains multiple sentences which answer the first and the third sub-questions, leaving the second question un-solved. For simplicity, we split the question into three questions and generate three question-answer pairs, as shown in Figure1. In principle, for QA matching in the formal text, there are two main challenges:

First, for a particular sub-question, the answer contains multiple sentences, some of which do not answer this sub-question but other sub-questions. For instance, in **E2**, for the first question "*Will the response time slow after updating os?*", only the first sentence answers this question and the fourth sentence answers the third question "*What about the screen?*".

Second, the answer even might contain many uncorrelated sentences. For instance, in **E2**, the sentence "*It has 4G RAM*" and "*But you must use it carefully*" are uncorrelated with all three sub-questions. That is to say, there exits noisy information in some answers in informal text.

In this paper, we focus the research on QA matching with informal text. First of all, we build a novel QA matching corpus with informal text which contains many question-answer pairs from a product reviewing website. Then, we attempt to propose a novel QA matching approach for informal text. To deal with the first challenge above, we propose a novel matching approach, namely, One vs. Many Matching, to learn the matching measurement with one question sentence and multiple answer sentences. Specifically, we first adopt the BIMPM (Wang et al., 2017) to implement the matching measurement with one question sentence and one answer sentence, namely One vs. One Matching. Then, we learn the One vs. Many Matching representation by integrating all One vs. One Matching representations.

Furthermore, to deal with the second challenge, we introduce both the word-level and sentence-level attention mechanisms. Specifically, the word-level attention is applied in the One vs. One Matching learning process while the sentence-level attention is applied in the One vs. Many Matching learning process.

The remainder of this paper is organized as follows. Section 2 discusses the related work on QA matching. Section 3 presents data collection and annotation. Section 4 proposes our approach to QA matching. Section 5 reports the experimental results. Finally, section 6 gives the conclusion and our future work.

2 Related Work

2.1 Corpus construction

In previous studies for researching QA matching, there are mainly two corpora, namely TREC-QA and WikiQA. Specifically, TREC-QA is proposed by Wang et al. (2007) where the questions are a mixture of questions from both query logs and human editors, and the answers are selected from documents returned by past participating teams in TREC-QA tracks. Each question has at least one answer. WikiQA is proposed by Yang et al (2015). where the questions are sampled from real queries of *Bing* without editorial revision and the answers are from the summary section of a Wikipedia page of the topic. Different from TREC-QA, WikiQA has two-thirds questions with no matching answers.

Unlike these two corpora, this paper proposes a novel corpus for QA matching research where the question-answer pairs are real ones from "*Asking People*" in *Taobao*. Moreover, different from the above corpora, the question-answer pairs in our corpus are informal text.

2.2 Matching methods

Generally speaking, QA matching methods could be split into two categories: shallow learning methods and deep learning methods.

In shallow learning methods, some shallow learning algorithms, such as CRF, SVM and MaxEnt, are employed to train the learning models (Wang et al., 2010). Besides the learning algorithms, the related studies on shallow learning methods mainly focus on feature engineering, using linguistic tools and using external resources, such as lexical semantic resources (Yih et al., 2013), tree edit distance (Yao and Durme, 2013) and named entities (Severyn and Moschitti, 2013).

[1] https://www.taobao.com/

In deep learning methods, some neural network algorithms are employed to train learning models. Briefly, these methods could be categorized into three categories, i.e., siamense networks, attentive networks and compare-aggregate networks. In siamense networks, related studies use classic neural networks, such as LSTM and CNN, to get the representations separately and then concatenate them to classify. (Feng et al., 2015; Yang et al., 2015; Bowman et al., 2015). In attentive networks, instead of using the final time step of LSTM to represent a sentence, related studies use the attention strategy to get the weight of overall time steps and then use the weight to represent the sentence. (Tan et al., 2016; Hermann et al., 2015, Yin et al., 2015). In compare-aggregate networks, related studies use different matching strategy to get relationships within words. (He and Lin, 2016; Wang et al., 2017; Wang and Jiang, 2016; Trischler et al., 2016; Parikn et al., 2016.).

However, all above approaches are similar to our One vs. One Matching model which deals with the matching measurement between one sentence (or one piece of text) and another sentence (or another piece of text). In contrast, our approach is a One vs. Many Matching model which deals with the matching measurement between one sentence (or one piece of text) and multiple sentences (or multiple pieces of text).

3 Data Collection and Annotation

We collect 4,060 question-answer pairs from "*Asking All*" in *Taobao,* which is the most famous and biggest electronic business platform in China. The question-answer pairs are mainly from the *electronic* domain. Note that if a question contains multiple sub-questions, we split the question into several sub-questions and each sub-question and the whole answer is considered as a question-answer pair. For instance, as shown in Figure 1, the question in Example 2 contains three sub-questions, and we split it into three question-answer pairs.

To guarantee a high annotation agreement, we propose some annotation guidelines after several times of annotation processes on a limited size of data. Then, we ask more people to annotate the whole data set according to these annotation guidelines.

Generally, the two categories, "*Matching*" and "*Non-matching*" has following cases.

1) About the "*Matching*" category

(a) The answer directly answers the question. **E3** is an example of this case.

E3: Q: *Is the response time slow?*
A: *Slow.*

(b) The answer indirectly answers the question. **E4** is an example of this case.

E4: Q: *Is the response time slow?*
A: *Smooth.*

(c) The answer conditionally answers the question. **E5** is an example of this case.

E5: Q: *How long can the battery take?*
A: *If you don't play games, it can last two days.*

2) About the "*Non-matching*" category

(d) The answer is irrelevant with answer. **E6** is an example of this case.

E6: Q: *Is the microphone clear?*
A: *The battery life is decent. I like it.*

(e) The answer is about the product but doesn't talk about the aspect in the question. **E7** is an example of this case.

E7: Q: *Is the screen clear?*
A: *The quality of the screen is very good. It is a good cell phone.*

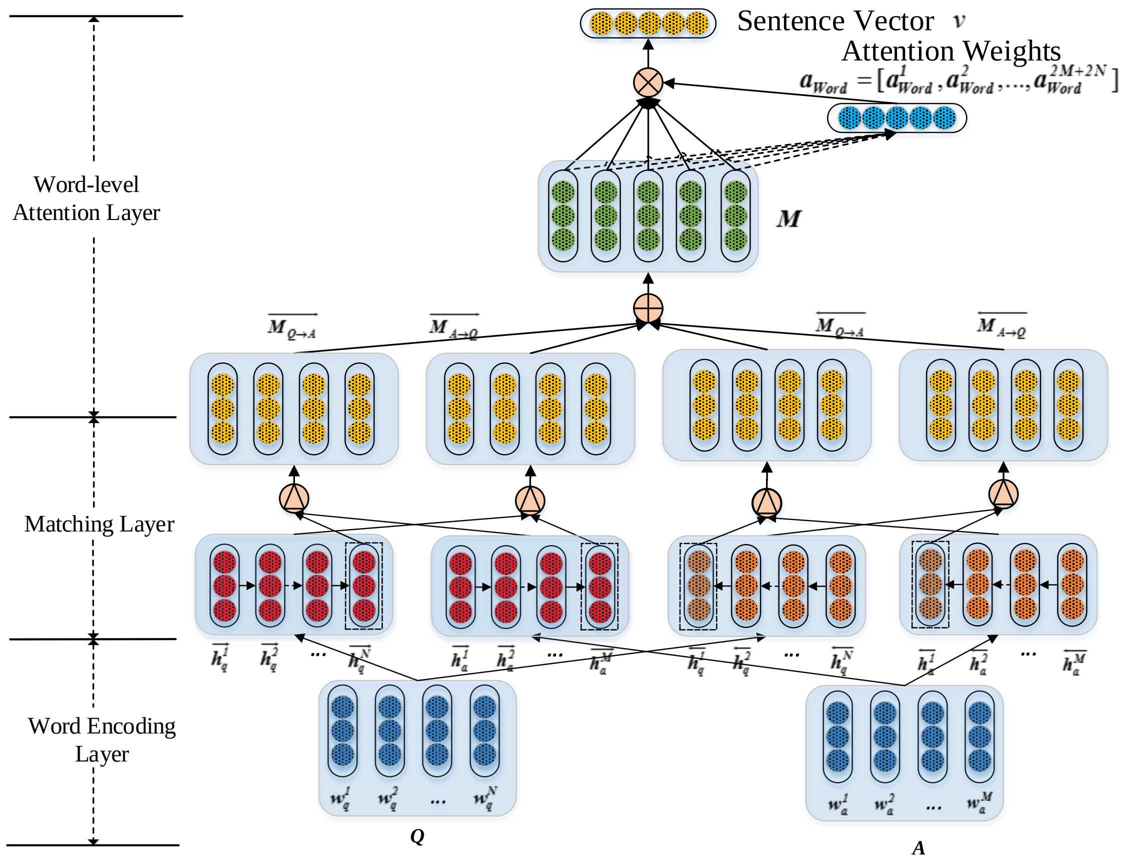
(f) The answer is an unknown response. **E8** is an example of this case.

E8: Q: *What about the performance of the phone?*
A: *I don't know, I bought it for my dad.*

For each question-answer pair, we assign two annotators to annotate the category, and the *Kappa* consistency check value of the annotation is 0.83. To deal with the question-answer pairs which are inconsistently annotated by two annotators, an expert is assigned to check them,

Category	Number
Matching	2505
Non-matching	1555

Table 1: Category distribution of the annotated data



Sentence Vector v
Attention Weights
$a_{Word} = [a^1_{Word}, a^2_{Word}, ..., a^{2M+2N}_{Word}]$
M
Word-level Attention Layer
Matching Layer
Word Encoding Layer
$\overline{M_{Q\to A}}$
$\overline{M_{A\to Q}}$
$\overrightarrow{h^1_q}$ $\overrightarrow{h^2_q}$ ... $\overrightarrow{h^N_q}$
$\overrightarrow{h^1_a}$ $\overrightarrow{h^2_a}$ ... $\overrightarrow{h^M_a}$
$\overleftarrow{h^1_q}$ $\overleftarrow{h^2_q}$ ... $\overleftarrow{h^N_q}$
$\overleftarrow{h^1_a}$ $\overleftarrow{h^2_a}$ ... $\overleftarrow{h^M_a}$
w^1_q w^2_q ... w^N_q
w^1_a w^2_a ... w^M_a
Q
A

Figure 2: One vs. One Matching model with word-level attention

ensuring the quality of data annotation. Table 1 shows the category distribution of the corpus.

4 Our Approach

In this section, we propose our approach to QA matching in two steps. First, we propose the One vs. One Matching model which measures the matching between one sentence of the question text and one sentence in the answer text. Second, we propose the One vs. Many Matching model which measures the matching between one sentence of the question text and all sentences in the answer text.

4.1 One vs. One Matching Model

Figure 2 shows the overall architecture of our One vs. One Matching approach to QA matching. The involved layers are introduced in detail as following.

Word Encoding Layer. This layer has two inputs: the whole question sentence and one answer sentence. We represent the two sentences with *d*-dimensional vectors which is pre-trained with word2vec[2]. Then we utilize two bi-directional LSTM to encode the both of the sequences into contextual embeddings for each time step of the question and the sub answer sentence.

$$[\overrightarrow{h^1_q}, \overrightarrow{h^2_q}, ..., \overrightarrow{h^N_q}] = \overrightarrow{LSTM}([w^1_q, w^2_q, ..., w^N_q]) \tag{1}$$

$$[\overrightarrow{h^1_a}, \overrightarrow{h^2_a}, ..., \overrightarrow{h^M_a}] = \overrightarrow{LSTM}([w^1_a, w^2_a, ..., w^M_a]) \tag{2}$$

$$[\overleftarrow{h^1_q}, \overleftarrow{h^2_q}, ..., \overleftarrow{h^N_q}] = \overleftarrow{LSTM}([w^1_q, w^2_q, ..., w^N_q]) \tag{3}$$

$$[\overleftarrow{h^1_a}, \overleftarrow{h^2_a}, ..., \overleftarrow{h^M_a}] = \overleftarrow{LSTM}([w^1_a, w^2_a, ..., w^M_a]) \tag{4}$$

Matching Layer. This is the core layer in our One vs. One matching model. The goal of this layer is to compare the final contextual embedding (in the final time step of LSTM) of the sentence in the question against all contextual embeddings (in all time steps of LSTM) of one sentence in the answer. As

[2] https://code.google.com/archive/p/word2vec/

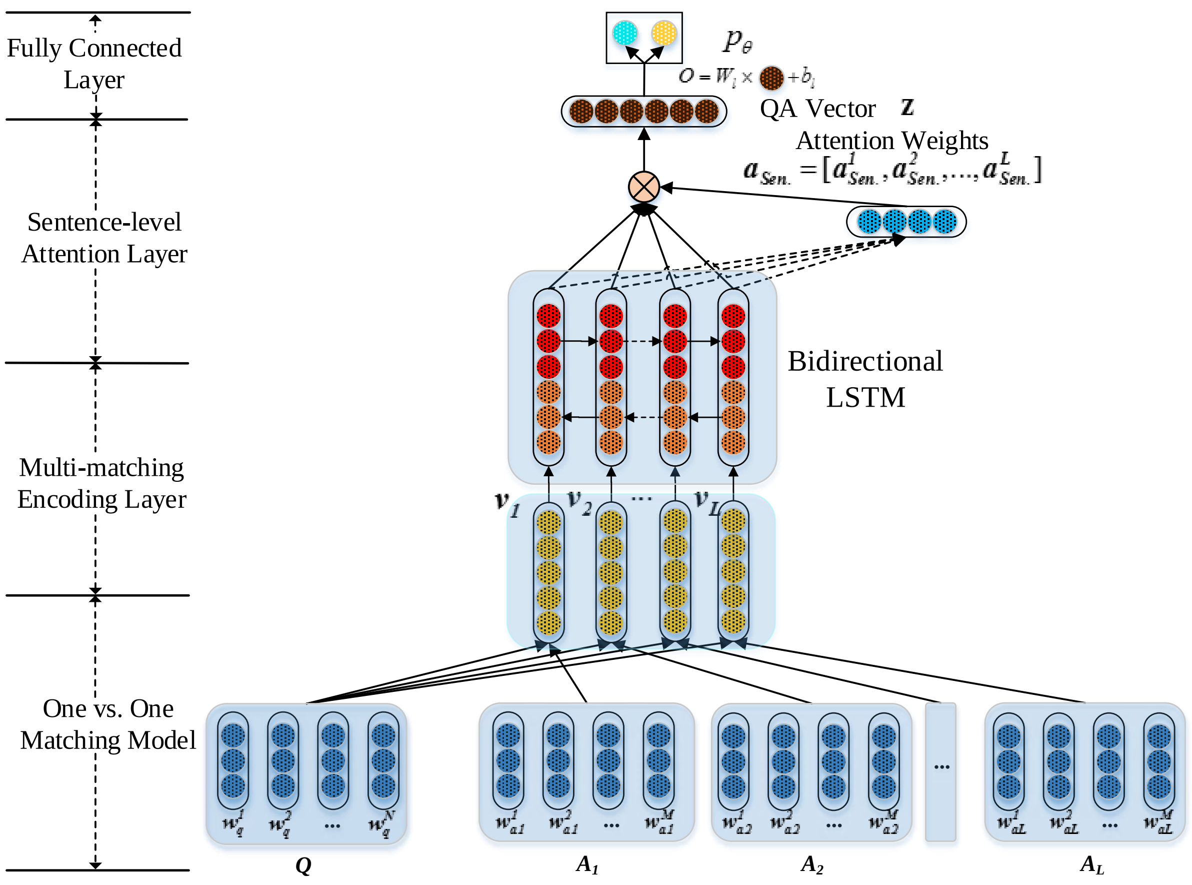

Fully Connected Layer
Sentence-level Attention Layer
Multi-matching Encoding Layer
One vs. One Matching Model
p_θ
$O = W_i \times \bullet + b_i$
QA Vector Z
Attention Weights
$a_{Sen.} = [a^1_{Sen.}, a^2_{Sen.}, \dots, a^L_{Sen.}]$
Bidirectional LSTM
v_1
v_2
v_L
Q
A_1
A_2
A_L

Figure 3: One vs. Many Matching model with sentence-level attention

shown in Figure 2, we match two sentences Q and A in two directions.

Formally, we use cosine function to calculate the similarity between two vectors, i.e.,

$$\overrightarrow{s_{q_N - a_i}} = cosine(\overrightarrow{h_q^N}, \overrightarrow{h_a^i}) \tag{5}$$

Inspired by Wang et al. (2017), we use the multi-perspective full matching strategy therein, i.e.,

$$\overrightarrow{s_{q_N - a_i}^k} = cosine(W^k \circ \overrightarrow{h_q^N}, W^k \circ \overrightarrow{h_a^i}) \tag{6}$$

where $\circ$ is the element-wise multiplication, and W^k is the k-th row of W, which controls the k-th perspective and assigns different weights to different dimensions of the dimension space.

Then, we obtain the similarity matrix $\overrightarrow{M_{Q \to A}} \in R^{P \times M}$, i.e.,

$$\overrightarrow{M_{Q \to A}} = \begin{bmatrix} \overrightarrow{s_{q_N - a_1}^1} & \cdots & \overrightarrow{s_{q_N - a_M}^1} \\ \vdots & \ddots & \vdots \\ \overrightarrow{s_{q_N - a_1}^P} & \cdots & \overrightarrow{s_{q_N - a_M}^P} \end{bmatrix} \tag{7}$$

where P means the number of perspectives.

Similarly, we get the matrices $\overleftarrow{M_{Q \to A}}$, $\overrightarrow{M_{A \to Q}}$ and $\overleftarrow{M_{A \to Q}}$.

Word-level Attention Layer. The goal of this layer is to assign weights to the similarity matrix and then get the most informative representation of the question sentence and one answer sentence.

First, we concatenate the $Q \to A$ and $A \to Q$ similarity matrix as the question-answer matrix.

$$M = \overrightarrow{M_{Q \to A}} \oplus \overleftarrow{M_{Q \to A}} \oplus \overrightarrow{M_{A \to Q}} \oplus \overleftarrow{M_{A \to Q}} \tag{8}$$

Second, we calculate the word attention weights a_{Word} with the following formula, i.e.,

$$a_{Word} = tanh(W_{Word} M + b_{Word}) \tag{9}$$

where W_{Word} is an intermediate matrix, b_{Word} is an offset value.

Third, we use a softmax function to normalize a_{Word}, i.e.,

$$a_{Word} = softmax(a_{Word}) \tag{10}$$

Finally, we multiply a_{Word} with M^T, and get the matching vector v, i.e.,

$$v = a_{Word} M^T \tag{11}$$

The matching vector v is the most informative representation of the question sentence and one answer sentence.

4.2 One vs. Many Matching Model

Figure 3 shows the overall architecture of our One vs. Many Matching approach to QA matching. The involved layers are introduced in detail as following.

One vs. One Matching Model. The purpose of this model is to use the question sentence and one answer sentence to calculate the most informative representation v_i. Suppose the answer contains L sentences, then we obtain the matching vectors between the question and answer, i.e., $V = [v_1, v_2, ..., v_L]$.

Multi-matching Encoding Layer. In this layer, we treat the above matching vectors as a sequence and employ a Bi-directional LSTM to learn a new representation, i.e.,

$$H = \overrightarrow{LSTM}(V) \oplus \overleftarrow{LSTM}(V) \tag{12}$$

Sentence-level Attention Layer. The goal of this layer is to get the most informative representation of the representation H.

First, we calculate the sentence attention weights $a_{Sen.}$, i.e.,

$$a_{Sen.} = tanh(W_{Sen.} H + b_{Sen.}) \tag{13}$$

where $W_{Sen.}$ is an intermediate matrix. $b_{Sen.}$ is an offset value.

Second, we use a softmax function to normalize $a_{Sen.}$, i.e.,

$$a_{Sen.} = softmax(a_{Sen.}) \tag{14}$$

Finally, we multiply $a_{Sen.}$ with H^T, and get the matching vector z, i.e.,

$$z = a_{Sen.} H^T \tag{15}$$

The matching vector z is the final representation for representing the matching measurement between the question and the whole answer.

Fully Connected Layer. The goal of this layer is to use the matching vector z to perform classification. Formally, we feed z to a softmax classifier, i.e.,

$$o = W_l z + b_l \tag{16}$$

where $o \in R^K$ is the output, W_l is the weight matrix and b_l is the bias. K is the number of categories. Then the probability of the category $k \in [1, K]$ is computed by:

$$p_\theta = \frac{exp(o_k)}{\sum_{t=1}^{K} exp(o_t)} \tag{17}$$

where θ denotes all parameters. Finally, the label with the highest probability stands for the predict label of the question-answer pair.

4.3 Model training

We use cross-entropy loss function to train our model end-to-end given a set of training data x_t, y_t, where x_t is the t-th question-answer pair to be predicted, and y_t is one-hot representation of the true category for x_t. We present this model as black-box function $\sigma(x)$ whose output is a vector representing the probability of categories. The goal of training is to minimize the loss function as following,

$$J(\theta) = -\sum_{t=1}^{N}\sum_{k=1}^{K} y_t^k \cdot \log\sigma(x_t) + \frac{l'}{2}\|\theta\|_2^2 \tag{18}$$

where N is the number of training samples and l' is a L_2 regularization to bias parameters.

We take Adam (Kingma and Ba, 2014) as the optimizing algorithm, and all the matrix and vector parameters are initialized with a uniform distribution in $\left[-\sqrt{6/(r+c)}, \sqrt{6/(r+c)}\right]$, where r and c are the rows and columns of the matrix (Glorot and Bengio, 2010).

5. Experiment

In this section, we systematically evaluate the performance of our approach to QA matching.

5.1 Experiment settings

Data Settings: As introduced in Section 3, the data contains 4,060 question-answer pairs and we randomly split the data into a training set (80% in each category), and a test set (the remaining 20% in each category). We also aside 10% data from training data as development data which is used to tune the parameters in each learning algorithm.

Word Segmentation and Embeddings: The Jieba[3] segmentation tool is employed to segment all Chinese text into words and word2vec[4] is employed to pretrain word embeddings using the data set that contains 200,000 question-answer pairs from the *electronic* domain. The dimensionality of the word vector is set to be 100 and the window size is set to be 1.

Sentence Split: We run sentence splitting with the CoreNLP[5] tool.

Hyper-parameters: The hyper-parameters values in the model are tuned according to performance in the development set. The size of units of the Bi-directional LSTM in the One vs. One Matching model is set to be 100, and the size of units of the Bi-directional LSTM in the One vs. Many Matching model is set to be 50. The default number of the sentences in an answer is set to be 5. The number of matching perspectives is set to be 20. The batch size is set to be 64.

Evaluation Measurement and Significance Test: The performance is evaluated using standard precision (*P*), recall (*R*), F-score (*F*) and *accuracy*. Furthermore, *t*-test is used to evaluate the significance of performance difference between two approaches.

5.2 Baselines

For comparison, we implement following approaches to QA matching:

- ➢ **LSTM:** A QA matching approach belonging to the siamense network, which is proposed by Bowman et al (2015). This approach employs a LSTM layer to encode the inputs.
- ➢ **Shallow Convolutional Neural Network (SCNN):** A state-of-the-art QA matching approach belonging to the siamense network, which is proposed by Zhang et al (2016). for the task of implicit discourse relation recognition.
- ➢ **Attentive LSTM:** A state-of-the-art QA matching approach belonging to an attentive network, which is proposed by Tan et al (2016).
- ➢ **MULT:** A state-of-the-art QA-matching approach belonging to the compare-aggregate network, which is proposed by Wang and Jiang (2017).
- ➢ **Bilateral Multi-Perspective Matching (BIMPM):** Another state-of-the-art QA matching approach belonging to the compare-aggregate network, which is proposed by Wang et al (2017). We use two implements where BIMPM (Full) employs the full matching strategy and BIMPM (Ensemble) employs all of the four matching strategies.

5.3 Our Approaches

Our approaches to QA matching are implemented with four different ways, i.e.,

[3] https://pypi.python.org/pypi/jieba/
[4] https://radimrehurek.com/gensim/models/word2vec.html
[5] https://stanfordnlp.github.io/CoreNLP/

	Macro-F	*Accuracy*
LSTM	0.600	0.649
SCNN	0.591	0.651
Attentive LSTM	0.632	0.697
MULT	0.651	0.691
BIMPM(Full)	0.654	0.700
BIMPM(Ensemble)	0.660	0.704
OMM	0.664	0.708
OMM+WA	0.674	0.713
OMM+SA	0.676	0.716
OMM+WA+SA	**0.689**	**0.721**

Table 2: Overall performances of different approaches to QA matching

	Matching			*Non-matching*		
	P	*R*	*F*	*P*	*R*	*F*
LSTM	0.647	0.864	0.740	0.656	0.356	0.461
SCNN	0.666	0.854	0.748	0.602	0.341	0.435
Attentive LSTM	**0.722**	0.863	0.786	0.611	0.394	0.479
MULT	0.720	0.824	0.769	0.618	0.471	0.534
BIMPM(Full)	0.712	0.860	0.779	0.663	0.442	0.531
BIMPM(Ensemble)	0.715	0.864	0.783	0.673	0.449	0.538
OMM	0.716	0.870	0.786	0.685	0.450	0.543
OMM+WA	0.709	0.883	0.786	0.725	0.460	0.563
OMM+SA	0.718	0.874	**0.789**	0.708	0.470	0.565
OMM+WA+SA	0.712	**0.883**	0.788	**0.744**	**0.488**	**0.590**

Table 3: Performances of different approaches to QA matching in each category

- **One vs. Many Matching (OMM):** This is the implementation where neither word-level attention layer nor the sentence-level attention layer is employed.
- **OMM with Word-level Attention (OMM+WA):** This is the implementation where only the word-level attention layer is employed.
- **OMM with Sentence-level Attention (OMM+SA):** This is the implementation where only the sentence-level attention layer is employed.
- **OMM with both Word-level and Sentence-level Attention (OMM+WA+SA):** This is the implementation where both the word-level attention layer and the sentence-level attention layer are employed.

5.4 Results

Table 2 and Table 3 show the overall and detailed performances of all approaches to QA matching. From this table, we can see that all our four approaches perform better than all baseline approaches and the significance test with *t*-test shows that the improvements are statistically significant (*p*-value<0.05). From the result, we can see that our approach is extremely superior in the category of *Non-matching*. For instance, our approach OMM+WA+SA outperforms the baseline approach LSTM with an improvement of 0.129 in terms of F-score. Among all our four approaches, OMM+WA+SA performs best, which highlights the importance of employing attention strategy in QA matching.

5.5 Visualization of Attention

In order to get a better understanding of the attention model and validate whether this model can capture the key information of the answer sentences, we visualize the word-level attention layer and sentence-level attention layer according to the obtained attention weights a_{Word} and $a_{Sen.}$.

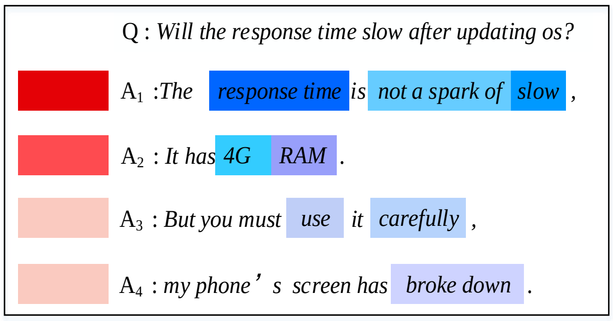

Q : Will the response time slow after updating os?
A1 :The response time is not a spark of slow ,
A2 : It has 4G RAM .
A3 : But you must use it carefully ,
A4 : my phone’ s screen has broke down .

Figure 4: The attention visualization for a question answer pair

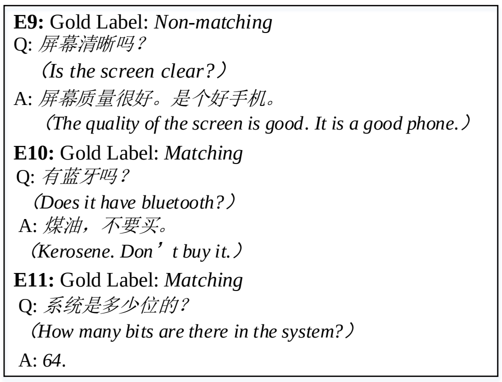

E9: Gold Label: *Non-matching*
Q: *屏幕清晰吗？*
(Is the screen clear?)
A: *屏幕质量很好。是个好手机。*
(The quality of the screen is good. It is a good phone.)
E10: Gold Label: *Matching*
Q: *有蓝牙吗？*
(Does it have bluetooth?)
A: *煤油，不要买。*
(Kerosene. Don’ t buy it.)
E11: Gold Label: *Matching*
Q: *系统是多少位的？*
(How many bits are there in the system?)
A: *64.*

Figure 5: Error Analysis

Figure 4 shows the attention visualizations for a question-answer pair. We normalize the word weight by the sentence weight to make sure that only informative words in informative sentences corresponding to the question are emphasized. In Figure 4, each line contains one sentence in an answer. Red denotes the sentence weight and blue denotes the word weight. The color depth indicates the importance degree of the attention weight for the question.

From the figure, we can see that the sentence-level attention function can select the informative sentences corresponding to the question, such as the first and the second sentences. In addition, the word-level attention function can select both the words and multi-word phrases carrying strong matching signals corresponding to the question, such as “*response time*”, “*not a spark of*”, and “*4G*”.

5.6 Error Analysis

Through analyzing the classification results of our approach, we find that QA matching is still challenging and some kinds of errors are discussed as following.

First, the question and the answer share the same aspect of the product but the concerned contents of the aspect are different. For instance, in Figure 5 and **E9**, the question and the answer both have the word “*screen*”. However, the question asks the clearness about the screen while the answer talks about the quality of the screen. Second, some words in the answer text have spelling mistakes. For instance, in Figure 5 and **E10**, the answer contains the word “*煤油* (*Kerosene*)” is a spelling mistake. It should actually be the word “*No*”, although their pronunciations in Chinese are similar. Third, some question-answer pairs contain the answer sentence which is too short, like **E11** in Figure 5.

6. Conclusion

In this paper, we propose a new corpus for the research on QA matching in informal text. Moreover, we propose a novel approach to QA matching, namely One vs. Many Matching, to handle the difficulty in which the answer contains multiple sentences. Furthermore, we employ both the word-level and sentence-level attentions in our approach to further improve the matching performance. Empirical studies show that the proposed approach performs significantly better than several strong baseline approaches.

In our future work, we would like to enlarge the scale of the corpus by annotating some data in other domains. Furthermore, we would like to evaluate the effectiveness of our approach to QA matching in some other domains or some other languages.

Acknowledgments

The research work is partially supported by two National Nature Science Foundation of China (No.61751206 and No.61672366), the National Key R&D Program of China under Grand No.2017YFB1002104 and Collaborative Innovation Center of Novel Software Technology and Industrialization. This work is also supported by the joint research project of Alibaba and Soochow University.

References

Samuel R. Bowman, Gabor Angeli, Christopher Potts and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing.* Association for Computational Linguistics, pages 632-642.

Minwei Feng, Bing Xiang, Michael R. Glass, Lidan Wang and Bowen Zhou. 2015. Applying Deep Learning to Answer Selection: A Study and An Open Task. *arXiv preprint arXiv:1508.01585.*

Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics*, pages 249-256.

Karl Moritz Hermann, Tomáš Kočiský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman and Phil Blunsom. 2015. Teaching Machines to Read and Comprehension. *arXiv preprint arXiv:1506.03340.*

Diederik P. Kingma and Jimmy Lei Ba. 2015. Adam: A Method for Stochastic Optimization. *arXiv preprint arXiv:1412.6980.*

Hua He and Jimmy Lin. 2016. Pairwise Word Interaction Modeling with Deep Neural Networks for Semantic Similarity Measurement. In *Proceedings of NAACL-HLT 2016*. Association for Computational Linguistics, pages 937–948.

Ankur P. Parikh, Oscar Täckström, Dipanjan Das and Jakob Uszkoreit. 2016. A Decomposable Attention Model for Natural Language Inference. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing.* Association for Computational Linguistics, pages 2249-2255.

Aliaksei Severyn and Alessandro Moschitti. 2013. Automatic Feature Engineering for Answer Selection and Extraction. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing.* Association for Computational Linguistics, pages 458-467.

Ming Tan, Cicero dos Santos, Bing Xiang and Bowen Zhou. 2016. Improved Representation Learning for Question Answer Matching. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics.* Association for Computational Linguistics, pages 464-473.

Adam Trischler, Zheng Ye, Xingdi Yuan, Jing He, Philip Bachman and Kaheer Suleman. 2016. A Parallel-Hierarchical Model for Machine Comprehension on Sparse Data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, pages 432-441.

Shuohang Wang and Jing Jiang. 2016. A Compare-Aggregate Model for Matching Test Sequences. *arXiv preprint arXiv:1611.01747.*

Mengqiu Wang and Christopher D. Manning. 2010. Probabilistic Tree-Edit Models with Structured Latent Variables for Textual Entailment and Question Answering. In *Proceedings of the 23rd International Conference on Computational Linguistics*. Association for Computational Linguistics, pages 1164-1172.

Zhiguo Wang, Wael Hamza and Radu Florian. 2017. Bilateral Multi-Perspective Matching for Natural Language Sentences. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pages 4144-4150.

Bingning Wang, Kang Liu and Jun Zhao. 2016. Inner Attention based Recurrent Neural Networks for Answer Selection. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics.* Association for Computational Linguistics, pages 1288-1297.

Yi Yang, Wen-tau Yih and Christopher Meek. 2015. WikiQA: A Challenge Dataset for Open-domain Question Answering. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing.* Association for Computational Linguistics, pages 2013-2018.

Xuchen Yao and Benjamin Ban Dueme. 2013. Answer Extraction as Sequence Tagging with Tree Edit Distance. In *Proceedings of NAACL-HLT 2013*. Association for Computational Linguistics, pages 858-867.

Wen-tau Yih, Ming-Wei Chang, Christopher Meek and Andrzej Pastusiak. 2013. Question Answering Using Enhanced Lexical Semantic Models. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics.* Association for Computational Linguistics, pages 1744-1753.

Wenpeng Yin, Hinrich Schütze, Bing Xiang, Bowen Zhou. 2015. ABCNN: Attention-Based Convolutional Neural Network for Modeling Sentence Pairs. *arXiv preprint arXiv:1512.05193.*

Biao Zhang, Jinsong Su, Deyi Xiong, Yaojie Lu, Hong Duan and Junfeng Yao. 2015. Shallow Convolutional Neural Network for Implicit Discourse Relation Recognition. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing.* Association for Computational Linguistics, pages 2230-2235.