Many vs. Many Query Matching with Hierarchical BERT and Transformer

Yang Xu, Qiyuan Liu, Dong Zhang, Shoushan Li(✉), Guodong Zhou

School of Computer Science and Technology, Soochow University, China
xuyang_yxu@126.com, {qyliu, dzhang17}@stu.suda.edu.cn
{lishoushan, gdzhou}@suda.edu.cn

Abstract. Query matching is a fundamental task in the Natural Language Processing community. In this paper, we focus on an informal scenario where the query may consist of multiple sentences, namely query matching with informal text. On the basis, we first construct two datasets towards different domains. Then, we propose a novel query matching approach for informal text, namely Many vs. Many Matching with hierarchical BERT and transformer. First, we employ fine-tuned BERT (bidirectional encoder representation from transformers) to capture the pair-wise sentence matching representations. Second, we adopt the transformer to accept above all matching representations, which aims to enhance the pair-wise sentence matching vector. Third, we utilize soft attention to get the importance of each matching vector for final matching prediction. Empirical studies demonstrate the effectiveness of the proposed model to query matching with informal text.

Keywords: Query Matching, Informal Text, BERT, Transformer.

1 Introduction

Query matching is a task that determines whether a pair of queries expresses the same intention. For instance, in Table 1, queries in E1 point to the same intention "tell me how to register the mobile phone number", so the system can give the same feedback to both the queries. Query matching has important research value in many areas. The past few years have witnessed a huge exploding interest in the research on query matching, due to its widely-used applications, such as response selection in dialogue system[1] and relevance evaluation in passage ranking [2].

Most existing studies in recent years only focus on query matching with formal text[3], which is often treated as a sentence-level text matching task. In real applications, such as query matching in online communities and smart customer service systems, user queries often consist of multiple sentences. For instance, E2 in Table 1 is a pair of queries extracted from smart customer service log of a bank. The query text is informal where Q1 consists of three sentences, and Q2 consists of two sentences. The second and third sentences "I need a loan settlement certificate to buy a house. Can you provide it to me?" of Q1 are matched with the third sentence "Can you give me a loan settlement certificate if I pay off the loan in advance?" of Q2 in the same meaning of

asking for proof. However, the first sentence of Q1 and Q2 are not related to the match result. As we can see, the matching task between queries consisting of multiple sentences is very complicated. Matching relationships are often hidden between one or more sentences of the queries, conventional sentence-level matching models are difficult to effectively solve query matching with informal text. Therefore, it is very important and also challenging to propose an approach to efficiently solving query matching with informal text. In past two years, a few studies, such as Wang et al. [4] and Shen et al. [5] have realized this challenge and proposed some approaches for informal text matching.

Table 1. Some query pair examples with their matching labels.

E1:	query pair in formal text		
Q1:	*怎么注册手机号啊?* *(How to register the mobile phone number?)*	Q2:	*告诉我号码注册教程。* *(Tell me the number registration tutorial.)*
Label: *Matching*			
E2:	query pair in informal text		
Q1:	*我的贷款提前还了，我买房需要一份贷款结清证明，你能提供给我吗?* *(My loan has been paid in advance. I need a loan settlement certificate to buy a house. Can you provide it to me?)*	Q2:	*我之前申请过贷款，提前还清的话可以提供给我证明吗?* *(I have applied for a loan before. Can you give me a loan settlement certificate if I pay off the loan in advance?)*
Label: *Matching*			

On the one hand, all existing studies in query matching are carried out by adding various neural networks on word embedding. Due to the semantic complexity of query text and the limitations of training corpus size, the improvement of various critical performance indicators has become a bottleneck. More recently, the pre-trained language models, such as ELMo [6], OpenAI GPT [7], and BERT [8], have demonstrated their strong performance in semantic representation. Especially, BERT (bidirectional encoder representation from transformers) has achieved state-of-the-art results in multiple NLP tasks. Since the input representation of BERT can be a pair of sentences, we can convert query pair into a single sentence pair by connecting all sentences of query with informal text. Then we can use BERT for query pair with informal text.

On the other hand, simply using BERT for query with informal text like this does not greatly improve the matching performance [9]. This is mainly because simply splicing a query composed of multiple sentences into one sentence will cause it to lose lots of information, and the sentence unrelated to the matching relationship will become noise that affects the matching accuracy. It is important and also challenging to achieve benefits from BERT while preserving the raw structure information of the query.

In this paper, we focus the research on query matching with informal text. First of all, we screen the existing text matching datasets and extract the query pairs with

informal texts, and finally form two datasets, one of them is in the financial domain and the other in the general domain.

To deal with the first challenge above, we propose a hierarchical query matching approach, namely, Many vs. Many Matching. First, in sentence-level, for each query pair such as $[queryA, queryB]$, we segment both the $queryA$ and $queryB$ into sentence list. Then each element in one sentence list corresponds to each element in another to form a sequence $[[senA_1, senB_1], [senA_1, senB_2], [senA_2, senB_1], [senA_2, senB_2]]$. Then We can model each element with sentence-level matching model. Second, in text-level, we integrate sentence-level matching information from sentence pair sequence. Furthermore, to deal with the second challenge, we describe mvmBERT, a simple variant of BERT. It takes a sequence of text pairs as input, and for each text pair in the sequence, mvmBERT encodes it through a 12-layer BERT. The hidden state sequence obtained by the BERT is finally passed through an integration layer consisting of multiple layers of Transformers to obtain the output of the model. Finally, we use a simple attention layer to weight the output of the integration layer to get the high-level matching information.

2 Related Works

2.1 Query Matching Corpus

In the latest studies for query matching, there are mainly three related query matching datasets, namely CCKS[1] query matching dataset, ATEC[2] question matching dataset, and LCQMQ[3] (A Large-scale Chinese Question Matching Corpus) [10]. Specifically, CCKS query matching dataset is proposed by WeBank in CCKS2018 (China Conference on Knowledge Graph and Semantic Computing). All data in this dataset come from the original banking domain smart customer service logs, and have been screened and manually annotated. ATEC question matching dataset is proposed by ATEC, all data comes from the actual application scenarios of the ATEC financial brain. LCQMQ is proposed by Harbin Institute of Technology in COLING 2018 (The 27th International Conference on Computational Linguistics). Data in LCQMQ is collected in general domain. In order to better research the novel scenario we proposed in this paper, we extract query pair with informal text from the above three datasets to form a dataset that focuses on informal query matching.

2.2 Text Matching methods

In the recent years, deep learning methods for text matching could be categorized into three categories: Siamese networks, attentive networks and compare-aggregate networks. In Siamese networks, related study separately obtains the representations of text to be matched through the same network structure, such as LSTM and CNN. Then calculates the distances of the two representations to model the similarity of text pair[11].

---

1 http://www.ccks2018.cn

2 https://dc.cloud.alipay.com

3 http://icrc.hitsz.edu.cn/info/1037/1146.htm

In attentive networks, instead of using the final output of hidden state to represent a sentence, related studies use attention mechanism to learn the weight of each position in the sequence to the final representation of the sequence [12]. In compare-aggregate networks, related studies use different matching mechanisms to obtain comparison information at different levels in the sequence [13]. However, unlike the methods above, in this paper we use BERT to model the matching relationship between two sentences.

2.3 BERT- and Transformer-based Neural Networks

BERT, defines a Transformer-based network that uses a simple masked language model strategy trained on Wikipedia, substantially improving state-of-the-art models when fine-tuned on BERT's contextual embeddings. BERT can use a single text sequence or a pair of text sequences as input to the model and then output a deep coded representation of the input sequence. Due to the superior performance of BERT in various NLP tasks, many BERT-based studies for different downstream tasks have recently emerged. Zhang et al. [14] use BERT for text summarization and Sun et al. [9] use BERT for sentiment analysis. At the same time, some studies focus on analyzing the impact of the output of each level of the BERT on different tasks, such as Kondratyuk et al. [15]. However, in this paper, we try to add an integration layer composed of multiple layers of Transformers on the output layer of the BERT to summarize the sequence features.

Table 2. Some query pair examples in raw corpora.

E1: query pair with informal text			
Q1:	*我的贷款提前还了，我买房需要一份贷款结清证明，你能提供给我吗?* *(My loan has been paid in advance. I need a loan settlement certificate to buy a house. Can you provide it to me?)*	Q2:	*我之前申请过贷款，提前还清的话可以提供给我证明吗?* *(I have applied for a loan before. Can you give me a loan settlement certificate if I pay off the loan in advance?)*
Label: *Matching*			
E2: query pair with formal text			
Q1:	*怎么注册手机号啊?* *(How to register the mobile phone number?)*	Q2:	*告诉我号码注册教程。* *(Tell me the number registration tutorial.)*
Label: *Matching*			
E3: query pair with formal text			
Q1:	*你好，如何使用掌上银行?* *(Hello, how to use Pocket Bank?)*	Q2:	*能发给我掌上营业厅的安装包么？谢谢!* *(Can you send me the installation package for my handheld business hall? Thank you!)*
Label: *Non-matching*			

3 Data Collection

In this paper, we first construct two datasets in different domain for query matching with informal text. Our datasets are derived from the following three public datasets: CCKS query matching data set, ATEC question matching dataset, and LCQMQ (A Large-scale Chinese Question Matching Corpus). The data in CCKS query matching data set and ATEC question matching dataset are mainly in the financial domain, while data in LCQMQ are mainly in the general domain. We extract all the query pair with informal text from this three datasets, which means for each sample in the dataset, if each query in the query pair consists of more than one sentence, we will extract it and add it to our new dataset. For instance, as shown in Table2, E1 is a query pair with informal text while E2 is a query pair with formal text, as each query in E2 has only one sentence. Note that in the extraction process, we will filter out all the sentences that have high frequency but without actual meaning such as “你好” (hello), “请问一下” (excuse me) and “谢谢” (thank you). As is shown in Table2, each query in E3 contains two sentences, but it does not belong to our new dataset.

After extraction and proofreading work, we extracted a query matching dataset based on financial domain from CCKS query matching dataset and ATEC question matching dataset, namely Informal_Financial, which contains 36,000 query pairs with informal text. While we extract a query matching dataset based on general domain from LCQMQ, namely Informal_General, which contains 22,000 query pairs with informal text. The specific information of the two data sets is shown in the Table 3.

Table 3. Category distribution of the data set.

Dataset	Informal_Financial	Informal_General
Domain	Financial	General
Number of query pairs	36,000	22,000
Number of matching pairs	16,740	9,340
Number of non-matching pairs	19,260	12,660

4 Approach

In this section, we propose our Many vs. Many Matching approach to query with informal text in two steps. First, we propose the BERT-based sentence-level matching model which measures the matching between one sentence of the query text and one sentence in the other query. Then we use layer attention to combine the multi-layer output of BERT instead of only using the results of the last layer to enhanced output. Second, we propose the integration layer which consists of multiple layers of Transformer. Integration layer integrates the matching information for all sentence pairs obtained from the query pair and integrate. Finally, the integration result is input into the integration attention layer to get the final matching representation.

4.1 Pair-wise Sentence Matching Model based on fine-tuned BERT with Layer Attention

One of the BERT's pre-training tasks is the next sentence prediction task. Therefore, BERT can model the relationship of sentence pair and understand sentence relationships in the process of fine-tuning. To model a pair of sentences with BERT, we should treat the sentence pair into a specific form and use it as the input of the BERT, so we can simply get the first state output of the last layer in the BERT as the sentence matching vector. Specifically, we insert a [CLS] token in the first position of the sentence pair and a [SEP] token after each sentence, as is shown below.

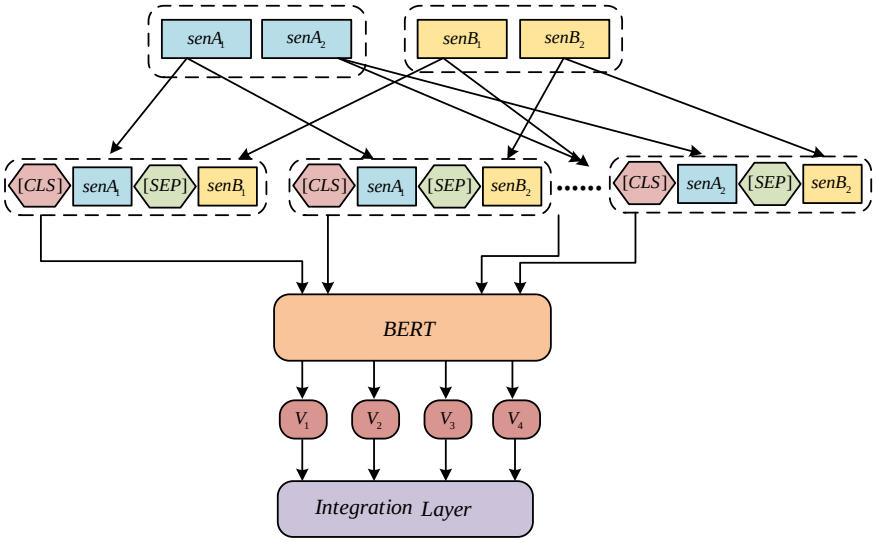

Fig. 1. Our Many vs. Many Matching approach based on BERT

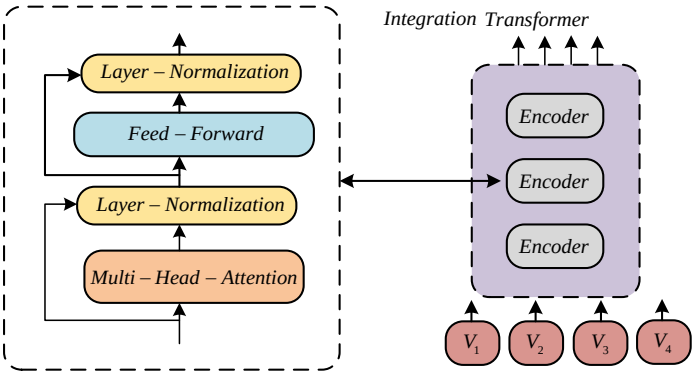

Fig. 2. The architecture of Integration Layer based on Transformer

The [CLS] is used as a symbol to aggregate features from a pair of sentences. For example, a pair of sentence: "*给我办理贷款的教程 & 怎么办理贷款*" *(Give me a loan tutorial & How to apply for a loan)*:

$$Input = [CLS]Give\ me\ a\ loan\ tutorial[SEP]How\ to\ apply\ for\ a\ loan[SEP] \quad (1)$$

$$PairVec = BERT\_sequenceoutput(Input) \quad (2)$$

However, studies suggest that when using BERT, combining the output of the last several layers instead of only using the last layer is more beneficial for the downstream tasks. So in this section we propose layer attention to combine the output of the last several layers in BERT. Specifically,

$$u_i = \tanh(W_w e_i + b_w) \quad (3)$$

$$\alpha_i = \frac{\exp(u_i^{\mathrm{T}} \cdot u_w)}{\sum_i \exp(u_i^{\mathrm{T}} \cdot u_w)} \quad (4)$$

We feed the last *i*-th layer output of BERT, e_i through a one-layer MLP to get its hidden representation u_i , then we measure the importance of e_i based on the similarity of a randomly initialized vector u_w and its hidden representation u_i . After getting a normalized importance weight α_i through a softmax function, we get the pair representation, combining the output of the last several layers as E :

$$V = \sum_i \alpha_i \cdot e_i \quad (5)$$

4.2 Many vs. Many Matching Model based on BERT for Query Pair

In the last section, we can get the pair representation using BERT. However, each query with informal text has more than one sentence, BERT cannot handle the matching of multiple sentences with multiple sentences, therefore, we describe mvmBERT, a simple variant of BERT. As is shown in Figure3, mvmBERT takes a sequence of text pairs as input, and for each sentence pair in the sequence, mvmBERT encodes it through a 12-layer baseBERT. Then hidden state sequence $[V_1, V_2, ..., V_{N\times M}]$ is passed through an integration layer, we can add a simple attention layer to get the final high-level match representation.

For example, a query pair with informal text $[queryA, queryB]$, assuming that $queryA$ consists of N sentences and $queryB$ has M sentences. We first segment both the $queryA$ and $queryB$ into sentence list:

$$queryA = [senA_1, senA_2, ..., senA_N] \quad (6)$$

$$queryB = [senB_1, senB_2, ..., senB_M] \quad (7)$$

Then we pair each sentence in $queryA$ with each sentence in $queryB$. Through this operation, we get a sentence pair sequence of length $N \times M$:

$$[[senA_1, senB_1], [senA_1, senB_2], ..., [senA_i, senB_j], ..., [senA_N, senB_M]] \quad (8)$$

Then, using sentence pair matching model based on BERT with layer attention:

$$V_{ij} = LayerAttention(BERT(senA_i, senB_j))_h \quad (9)$$

where *h* indicates that only the last *h* -layer output of the last BERT is considered in the calculation. Through the above calculations, the original query pair has become a vector sequence $[V_1, V_2, ..., V_{N\times M}]$ which is encoded by BERT. Now we consider two possible integration layer structures: recurrent neural network and Transformer.

4.3 Integration Layer based on Recurrent Neural Network

RNN is the most commonly used model for processing sequence data. In this section, we use BiGRU with attention to process the vector sequence, for each time step:

$$h_t = BiGRU(V_t) \tag{10}$$

Then we can receive final high-level match representation through the attention layer:

$$M = \sum_t \beta_t \cdot h_t \tag{11}$$

Where β_t is obtained by the same attention mechanism as above, and M is the final high-level matching representation**.** The final label probability is obtained from a simple classification layer:

$$p_{matching} = softmax(W_m \cdot M + b_m) \tag{12}$$

4.4 Integration Layer based on Transformer

Instead of BiGRU, Integration layer based on Transformer uses a pure attention structure. Research shows that Transformer has stronger feature extraction capabilities than RNN in many tasks. As is shown in Figure4, Transformer extracts the features of the vector sequence obtained by BERT:

$$\tilde{T}^l = LN(T^{l-1} + MultiHATT(T^{l-1})) \tag{13}$$

$$T^l = LN(\tilde{T}^l + FeedForward(\tilde{T}^l)) \tag{14}$$

Where $T^0 = PosEmbedding(V)$ and V is the vector obtained by BERT, *PosEmbedding* maps the positional information of V to a vector representation of a fixed dimension. In this section, we randomly initialize the positional embedding matrix so that it can be trained, just like original BERT paper. *MultiHATT* is the multi-head attention operation and *FeedForward* is a simple feedforward neural network, while l indicates the number of transformer layers that make up the integration layer. The final label probability is obtained from a simple attention layer and a classification layer:

$$M = \sum_t \beta_t \cdot T_{lt} \tag{15}$$

$$p_{matching} = softmax(W_m \cdot M + b_m) \tag{16}$$

5 Experiment

In this section, we systematically evaluate the performance of our Many vs. Many Matching approach based on BERT.

5.1 Experiment Settings

- **Data Settings:** As introduced in Section 3, we extract two datasets from the

existing three datasets, one based on the financial domain, namely Informal_Financial, and another based on the general domain, namely Informal_General. Informal_Financial contains 36,000 query pairs with informal text, and Informal_General contains 22,000 query pairs with informal text. For each data set, 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 set which is used to tune the parameters.

- **Word Segmentation and Sentence Split:** The Jieba[4] segmentation tool is employed to segment all Chinese text into words. Word2vec[5] is employed to pretrain word embeddings, while the dimensionality of the word vector is set to be 300 and the window size is set to be 1. We run sentence splitting with the CoreNLP[6] tool.
- **Hyper-parameters:** The BERT version is BERT-Base, Chinese, which is pretrained on Chinese Simplified and Traditional. It has 12 Transformer layer, 768-hidden_size, 12 heads of multihead-attention, consisting of 110M parameters. The hyper-parameters values in the model are tuned according to performance in the development set. The hidden state size of BiGRU and Transformer are both 768. The batch size is set to be 64 and the max length of sequence is set to 40 while training the model.
- **Evaluation Metric:** We use *Macro-F1 (F)* and *Accuracy* to measure the divergences between predicted labels and ground-truth labels, where $F = \frac{2PR}{P+R}$ and the overall precision (*P*) and recall (*R*) are averaged on the corresponding scores from each category.

5.2 Baselines Approaches

In this section, we provide selected baseline approaches for thorough comparison. In addition, we also implement some state-of-the-art approaches in query matching.

- **Siamese LSTM:** A text matching approach belonging to the Siamese network, which is proposed by Bowman [16]. This approach employs LSTM layer to encode the text and calculate two text encoding distances to determine if they match.
- **SCNN:** A state-of-the-art text matching approach belonging to the Siamese network, which is proposed by Zhang [14]. For the task of implicit discourse relation recognition
- **Attentive LSTM:** A state-of-the-art text matching approach belonging to an attentive network, which is proposed by Tan [17].
- **MULT:** A state-of-the-art text-matching approach belonging to the compare-aggregate network, which is proposed by Wang [18].
- **BIMPM:** Another state-of-the-art text matching approach belonging to the compare-aggregate network, which is proposed by Wang [4].
- **Sentence BERT:** Splicing the query pair with informal text into sentence pair, using fine-tune BERT model to classify the sentence pair.

---

4 https://pypi.python.org/pypi/jieba/

5 https://radimrehurek.com/gensim/models/word2vec.html

6 https://stanfordnlp.github.io/CoreNLP/

5.3 Our Approaches

Our approaches to query matching are implemented with four different ways:

- **mvmBERT with RNN Integration (MVMR):** This is the implementation where we use BiGRU for integration layer and only use the last layer of BERT for the sentence pair modeling.
- **mvmBERT with Transformer Integration (MVMT):** This is the implementation where we use transformer for integration layer and only use the last layer of BERT for the sentence pair modeling.
- **mvmBERT with RNN Integration and Layer Attention (MVMR+LA):** This is the implementation where we use BiGRU for integration layer and use layer attention in last several layer of BERT for the sentence pair modeling.
- **mvmBERT with Transformer Integration and Layer Attention (MVMT+LA):** This is the implementation where we use transformer for integration layer and use layer attention in last several layer of BERT for the sentence pair modeling.

Table 4. Performance comparison of different approaches to query matching.

	Informal_Financial		Informal_General	
	Accuracy	*Macro-F1*	*Accuracy*	*Macro-F1*
Siamese LSTM	0.6990	0.7010	0.7122	0.7123
SCNN	0.6870	0.6941	0.7116	0.7120
Attentive LSTM	0.7115	0.7237	0.7254	0.7300
MULT	0.7120	0.7122	0.7155	0.7157
BIMPM	0.7344	0.7380	0.7528	0.7598
Sentence BERT	0.7797	0.7991	0.8009	0.8013
MVMR	0.8001	0.8065	0.8232	0.8232
MVMT	0.8133	0.8139	0.8455	0.8459
MVMR+LA	0.8182	**0.8216**	0.8513	0.8516
MVMT+LA	**0.8207**	0.8214	**0.8580**	**0.8581**

Table 5. Performance of MVMT with different number of transformer.

	Informal_Financial		Informal_General	
	Accuracy	*Macro-F1*	*Accuracy*	*Macro-F1*
MVMT1	0.7945	0.7935	0.8018	0.8141
MVMT2	0.8133	0.8139	0.8455	0.8459
MVMT3	0.8005	0.8000	0.8283	0.8276
MVMT1+LA8	0.8004	0.8008	0.8362	0.8367
MVMT2+LA8	**0.8207**	**0.8214**	**0.8580**	**0.8581**
MVMT3+LA8	0.8129	0.8133	0.8435	0.8437

Table 6. Performance of MVMT with different number of attention layer.

	Informal_Financial		Informal_General	
	Accuracy	*Macro-F1*	*Accuracy*	*Macro-F1*
MVMR+LA6	0.8180	0.8183	0.8509	0.8515
MVMR+LA8	0.8182	**0.8216**	0.8513	0.8516
MVMR+LA12	0.8133	0.8135	0.8512	0.8511
MVMT2+LA6	0.8200	0.8204	0.8513	0.8516
MVMT2+LA8	**0.8207**	0.8214	**0.8580**	**0.8581**
MVMT2+LA12	0.8192	0.8189	0.8567	0.8569

5.4 Results

Table 4 show the overall and performances of all approaches to query matching. From this table, we can see that all our four approaches perform better than all baseline approaches. Among our four approaches, mvmBERT with Transformer Integration and Layer Attention has made the best performance, which proves the importance of layer attention and transformer-based integration layer.

Specifically, we also studied the effect of the number of transformers in the integration layer and the number of the layers in layer attention on the performance of the model. As is shown in Table4 and Table 5, wo can find that last 8 layers in layer attention and 2 transformers in integration layer is the best choice.

6 Conclusion

In this paper, we first construct two datasets based on different domains for query matching with informal text. Then we propose a novel approach to query matching with informal text, namely Many vs. Many Matching. Furthermore, we improve our matching approach by employing BERT to implement the matching measurement and adding an integration layer consisting of multiple layers of transformers on BERT to integrate the matching result. 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 collecting more data in more domains. Also, we would like to evaluate the effectiveness of our approach to query matching in some other domains or some other languages.

Acknowledgments

The research work is partially supported by the Key Project of NSFC No.61702149 and two NSFC grants No.61672366, No.61673290.

References

1. Zhou, X., Li, L., Dong, D., Liu, Y., Chen, Y., Zhao, W.X., Yu, D., Wu, H.: Multi-turn response selection for chatbots with deep attention matching network. In: Proceedings of the

56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 1118–1127 (2018)
2. Qiao, Y., Xiong, C., Liu, Z., Liu, Z.: Understanding the behaviors of bert in ranking. arXiv preprint arXiv:1904.07531 (2019)
3. Severyn A, Moschitti A. Learning to rank short text pairs with convolutional deep neural networks[C]//Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval. ACM, 2015: 373-382
4. Wang, L., Li, S., Sun, C., Si, L., Liu, X., Zhang, M., Zhou, G.: One vs. many qa matching with both word-level and sentence-level attention network. In: Pro-ceedings of the 27th International Conference on Computational Linguistics. pp. 2540–2550 (2018)
5. Shen, C., Sun, C., Wang, J., Kang, Y., Li, S., Liu, X., Si, L., Zhang, M., Zhou, G.: Sentiment classification towards question-answering with hierarchical matching network. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. pp. 3654–3663 (2018)
6. Peters, M.E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., Zettlemoyer, L.: Deep contextualized word representations. arXiv preprint arXiv:1802.05365 (2018)
7. Radford, A., Narasimhan, K., Salimans, T., Sutskever, I.: Improving lan-guage understanding by generative pre-training. URL https://s3-us-west-2. ama-zonaws. com/openai-assets/research-covers/languageunsupervised/language un-derstanding paper. pdf (2018)
8. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirec-tional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
9. Sun, C., Huang, L., Qiu, X.: Utilizing bert for aspect-based sentiment analysis via constructing auxiliary sentence. arXiv preprint arXiv:1903.09588 (2019)
10. Liu, X., Chen, Q., Deng, C., Zeng, H., Chen, J., Li, D., Tang, B.: Lcqmc: A large-scale chinese question matching corpus. In: Proceedings of the 27th International Conference on Computational Linguistics. pp. 1952–1962 (2018)
11. Feng, M., Xiang, B., Glass, M.R., Wang, L., Zhou, B.: Applying deep learning to answer selection: A study and an open task. In: 2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU). pp. 813–820. IEEE (2015)
12. Yin, W., Sch¨utze, H., Xiang, B., Zhou, B.: Abcnn: Attention-based convolutional neural network for modeling sentence pairs. Transactions of the Association for Computational Linguistics 4, 259–272 (2016)
13. He, H., Lin, J.: Pairwise word interaction modeling with deep neural networks for semantic similarity measurement. In: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. pp. 937–948 (2016)
14. Zhang, H., Gong, Y., Yan, Y., Duan, N., Xu, J., Wang, J., Gong, M., Zhou, M.: Pretraining-based natural language generation for text summarization. arXiv preprint arXiv:1902.09243 (2019)
15. Kondratyuk, D.: 75 languages, 1 model: Parsing universal dependencies universally. arXiv preprint arXiv:1904.02099 (2019)
16. Bowman, S.R., Angeli, G., Potts, C., Manning, C.D.: A large annotated corpus for learning natural language inference. arXiv preprint arXiv:1508.05326 (2015)
17. Tan, M., Dos Santos, C., Xiang, B., Zhou, B.: Improved representation learning for question answer matching. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). vol. 1, pp. 464–473 (2016)
18. Wang, S., Jiang, J.: A compare-aggregate model for matching text sequences. arXiv preprint arXiv:1611.01747 (2016)