DUT-BIM at MEDIQA 2019: Utilizing Transformer Network and Medical Domain-Specific Contextualized Representations for Question Answering

Huiwei Zhou, Bizun Lei, Zhe Liu, Zhuang Liu
School of Computer Science and Technology
Dalian University of Technology
116024 Dalian, China
zhouhuiwei@dlut.edu.cn
{leibzun, njjnlz, zhuangliu1992}@mail.dlut.edu.cn

Abstract

In medical domain, given a medical question, it is difficult to manually select the most relevant information from a large number of search results. BioNLP 2019 proposes Question Answering (QA) task, which encourages the use of text mining technology to automatically judge whether a search result is an answer to the medical question. The main challenge of QA task is how to mine the semantic relation between question and answer. We propose BioBERT Transformer model to tackle this challenge, which applies Transformers to extract semantic relation between different words in questions and answers. Furthermore, BioBERT is utilized to encode medical domain-specific contextualized word representations. Our method has reached the accuracy of 76.24% and spearman of 17.12% on the BioNLP 2019 QA task.

1 Introduction

In medical field, the professional vocabulary is large and the semantics are complex, which makes manually selecting answers to a medical question from search results time consuming. The question answering (QA) task proposed by BioNLP 2019 (BEN ABACHA et al., 2019) aims to automatically extract answers to a medical question by using text mining technology. This task consists of two objectives: one is to determine whether each candidate answer can be used as the correct answer to a question, and the other is to rank the retrieved answers according to the relevance to a question.

The nature of QA task is to match the meaning rather than only match words between question and answer sentences. Several QA approaches based on syntax information have been developed to match the meaning between question and answer. Wang et.al. (2007) propose a statistical syntax-based model that softly aligns a question sentence with a candidate answer sentence. Tymoshenko and Moschitti (2015) encode semantic knowledge directly into syntactic tree representations of a pair of questions and answers for answers ranking. However, all these models rely on dependency parsers, suffering from error propagation.

Neural network-based methods can automatically learn the inherent semantic features and have achieved good performance on QA task. Wang and Nyberg (2017) employ an attentional encoder-decoder model based on long short-term memory (LSTM) (Hochreiter and Schmidhuber, 1997) for answer ranking, and their model achieves the best performance of 63.7% average score on the TREC LiveQA 2017 challenge (Agichtein et al., 2017). Yang et al. (2017) use a convolutional neural network (CNN) model to classify a question into a restricted set of 10 question types and crawl relevant online web pages to find the answers. However, all the models described above neglect the long range dependency between words in question and answer, limiting their capacity when question and answer sequences are long.

Transformer (Vaswani et al., 2017) is a model based entirely on attention mechanisms and has achieved success on several natural language processing (NLP) tasks, such as machine translation (Vaswani et al., 2017) and language understanding (Devlin et al., 2018). Transformer uses multi-head attention mechanisms to effectively capture the long-range dependency

Proceedings of the BioNLP 2019 workshop, pages 446–452
Florence, Italy, August 1, 2019.

information in context sequences, which is vital for question answering task.

Recently, language models (LM) based on large-scale corpus pre-training have made great progress in several NLP tasks, such as machine translation and natural language inference (NLI). ELMo (Peters et al., 2018) learns two unidirectional LMs based on LSTM networks which is able to capture both sub-word information and contextual clues. OpenAI GPT (Radford et al., 2018) uses a left-to-right Transformer (Vaswani et al., 2017), which introduces minimal task-specific parameters and is trained on the downstream tasks by simply fine-tuning the pre-trained parameters. The major limitation of pre-trained model above is that they are unidirectional, which limits the choice of architectures that can be used during pre-training. BERT (Devlin et al., 2018) employs a bidirectional Transformer encoder to fuse both the left and the right context and can explicitly model the relationship of a pair of text. Thus, it can make progress in paired NLP tasks, such as NLI and QA. Based on the BERT architecture, BioBERT (Lee et al., 2019) is a domain-specific language representation model pre-trained on large-scale biomedical corpora and effectively transfers the knowledge from biomedical texts to biomedical text mining models.

Corpus of QA task proposed by BioNLP 2019 contains answers with long text, which requires models to capture the long range dependency information across words in both question and answer sentences. Thus, we propose BioBERT Transformer (BBERT-T) model based on Transformer to model the associations between question and answer. Specifically, question and answer sequences are first passed to BioBERT to generate medical domain-specific contextualized representations. Then, the question and answer representations are fed into two Transformers, respectively, to capture the long range dependency information and semantic relation between question and answer. Finally, a weighted cross entropy loss is applied to further improve the performance. Our method achieves accuracy of 76.24% and spearman of 17.12% on the BioNLP 2019 QA task.

2 System Description

In MEDIQA2019 medical Question Answering (QA) task, given a question q and n_a candidate answers $\{a^1, a^2, ..., a^{n_a}\}$, we need build model to rank all candidate answers and to recognize correct answers to the question. Let T be the set of all the question-answer pairs. For each question-answer pair (q, a), we use BioBERT to encode the contextual information, which improves the model generalization capability. Then we propose two Transformers to learn the long range dependency information between words in question and answer, respectively. In this section, we introduce our approach for QA task in two steps: (1) the preprocessing of the corpus; (2) the structure of the model.

2.1 Preprocessing

Firstly, we lowercase all the questions and answers. Then, following (Fajcik et al., 2019), for each text of questions and answers, we use the tokenizer, that comes from Hugging Face PyTorch re-implementation of BERT[1], to split input words into most frequent *n*-grams in the pre-training corpus, effectively representing text at the sub-word level. Next, following (Vaswani et al., 2017), (q, a) pair sequences are truncated to have at most 300 tokens. At last, we use the truncated pair sequence $[q_1, q_2, ..., q_{l_q}, a_1, a_2, ..., a_{l_a}]$ as input, where l_q is the length of question, l_a is the length of a candidate answer and $l_q + l_a \leq 300$.

2.2 BioBERT Transformer Model (BBERT-T)

Structure of the proposed model is shown in Figure 1, which is composed of three layers: (1) BioBERT layer; (2) Transformer layer; (3) classification and ranking layer. Take the question-answer pair sequence $[q_1, q_2, ..., q_{l_q}, a_1, a_2, ..., a_{l_a}]$ as input to the BioBERT layer, achieving the question representation and answer representation. Then the two representations are fed to Transformer layer to extract the long range dependency information between words in question and answer, respectively. Finally, the outputs of Transformer layer are passed to a max pooling layer to generate features used to perform classification and ranking.

[1] https://github.com/huggingface/pytorch-pretrained-BERT

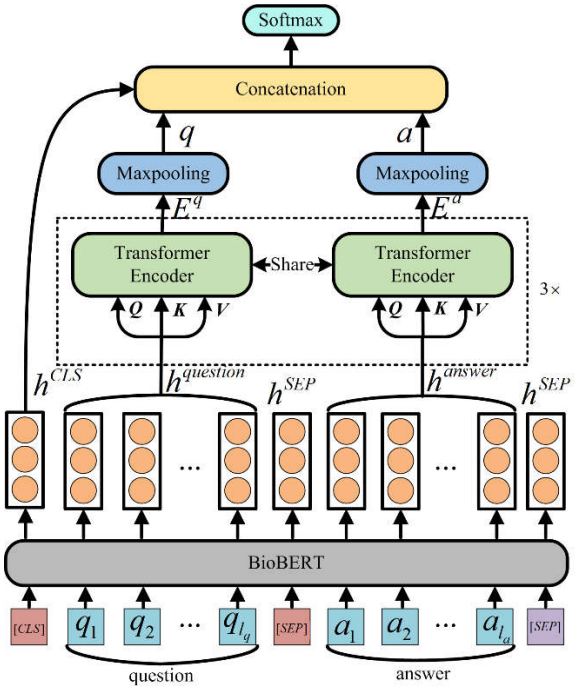

Figure 1: Architecture of BioBERT Transformer model.

The details of our model are described in the following subsections.

BioBERT layer: BioBERT (Lee et al., 2019) has achieved good performance after fine-tuning in several biomedicine NLP tasks. Therefore, we use the BioBERT to encode the question-answer pair sequence, which improves the model generalization capability. Following (Vaswani et al., 2017), given a question-answer pair sequence $[q_1, q_2, ..., q_{l_q}, a_1, a_2, ..., a_{l_a}]$, we add [CLS] token as the first token to the sequence and separate question and answer sequence with [SEP] token to get the input sequence, i.e. $[CLS, q_1, q_2, ..., q_{l_q}, SEP, a_1, a_2, ..., a_{l_a}, SEP]$. BioBERT is used to encode the input sequence and the final layer output is used as the contextualized representation of the question-answer pair $H = [h^{CLS}, h_1^q, ..., h_{l_q}^q, h^{SEP}, h_1^a, ..., h_{l_a}^a, h^{SEP}] \in \mathbb{R}^{n \times d}$, where $n = l_q + l_{a^i} + 3$ and d is the hidden layer dimension. Representations of questions and answers and the h^{CLS} will be used as inputs to the Transformer layer and the classification and ranking layer, respectively, which are described in following subsections. Note that all parameters of the BioBERT are fine-tuned during training.

Transformer layer: In this layer, two Transformers are applied to capture the long range dependency information and semantic relation between question and answer. To help better understand this layer, we provide a brief overview of Transformer (Vaswani et al., 2017).

The key component of Transformer is the multi-head attention layer that allows the model to jointly attend to information from different representation sub-spaces at different positions. Formally, given the queries Q, keys K, values V, multi-head attention builds upon scaled dot product attention mapping a query and a set of key-value pairs to an output:

$$Att(Q, K, V) = \text{soft max}(\frac{QK^T}{\sqrt{d}})V \quad (1)$$

where Q, K, V and output are all list of vectors with equal length, and d is the dimension size of K. Multi-head attention applies several scaled dot product attentions, which can be formulized as follows:

$$head_i = Att(W_i^Q Q, W_i^K K, W_i^V V) \quad (2)$$

$$\text{MultiHead}(Q,K,V) = W^H [head_1, head_2, \cdots, head_h] \quad (3)$$

where $W_i^Q \in \mathbb{R}^{d/h \times d}$, $W_i^K \in \mathbb{R}^{d/h \times d}$ and $W_i^V \in \mathbb{R}^{d/h \times d}$ are trainable parameter matrices and h represents the number of scaled dot product attention, or head. $[head_1, head_2, \cdots, head_h]$ is a concatenation of outputs of h heads. Note that, in this paper, we set $W^H \in \mathbb{R}^{d \times d}$ to a fixed identity matrix to reduce model complexity.

After multi-head attention layer, Transformer applies a two-layer full connection layer with ReLU activation:

$$H_q = \text{Re LU}(Q_{update} W_1 + b_1) W_2 + b_2 \quad (4)$$

where $W_1 \in \mathbb{R}^{d \times d}$, $W_2 \in \mathbb{R}^{d \times d}$, $b_1 \in \mathbb{R}^d$, $b_2 \in \mathbb{R}^d$ are trainable parameter matrices. $Q_{update} = \text{MultiHead}(Q,K,V)$ is the output of multi-head attention.

For both multi-head attention layer and the full connection layer, we use the residual concatenation (He et al., 2016) and Layer Normalization (Ba et al., 2016).

In our Transformer layer, we first split the contextualized representation $H = [h^{CLS}, h_1^q, ..., h_{l_q}^q, h^{SEP}, h_1^a, ..., h_{l_a}^a, h^{SEP}]$ from BioBERT into question representation

$H^q = [h_1^q, h_2^q, ..., h_{l_q}^q] \in \mathbb{R}^{l_q \times d}$ and answer representation $H^a = [h_1^a, h_2^a, ..., h_{l_a}^a] \in \mathbb{R}^{l_a \times d}$. Then question and answers representations are passed to two Transformers, respectively. Take the question for example, Question representation H^q forms the Q, K and V which are fed into a Transformer. The output of the Transformer for question is represented as E^q. In the same way, we can get E^a for answer. To establish the connection between question and answer, the two Transformers share the parameters. E^q and E^a will be used in the following classification and ranking layer.

2.3 Classification and Ranking Layer

In order to summarize the information of the questions and answers, we use the max pooling to generate question features $q \in \mathbb{R}^d$ and answer features $a \in \mathbb{R}^d$:

$$q = \max\text{pool}(E^q) \quad (5)$$

$$a = \max\text{pool}(E^a) \quad (6)$$

q and a are concatenated to form $[q, a]$ as the features for classification and ranking. To make full use of information about the relationship between question and answer, we further concatenate the classification embedding h^{CLS} from BioBERT layer to form the final features $[q, a, |q-a|, q \times a, h^{CLS}]$. Then, $[q, a, |q-a|, q \times a, h^{CLS}]$ is passed to a softmax layer to perform the classification. The softmax layer consists of a dense layer and a logistic regression classifier with a softmax function.

$$o = W_3[q, a, |q-a|, q \times a, h^{CLS}] + b_3 \quad (7)$$

$$p(y_t = j \mid T_t) = \text{soft}\max(W_o o + b_o) \quad (8)$$

where $W_3 \in \mathbb{R}^{5d \times d}$, $b_3 \in \mathbb{R}^d$, $W_o \in \mathbb{R}^{d \times 2}$ and $b_o \in \mathbb{R}^2$ trainable parameters, $j \in \{0,1\}$, and T_t represents tth training samples. We rank the answers according to the probability of being true answer (i.e. $p(y_t=1 \mid T_t)$).

In order to make full use of the reference score in the training set and to carefully control the loss, this paper applies a weighted cross entropy loss. For a candidate answer with the reference scores of 1, 2, 3, 4, the corresponding output labels should be 0, 0, 1, 1. We assign weights of 2, 1, 1, 2 to each label, respectively, when calculating the cross entropy loss:

$$\alpha = \begin{cases} 2, score = 1 \\ 1, score = 2 \\ 1, score = 3 \\ 2, score = 4 \end{cases} \quad (9)$$

$$loss = -\frac{1}{N}\sum_{t=1}^{N} \alpha \log p(y_t \mid T_t) \quad (10)$$

where N is the number of question-answer pairs.

3 Experiments

3.1 Dataset and Evaluation Metrics

Dataset: MEDIQA2019-Task3-QA task contains dataset of medical questions and the associated answers retrieved by CHiQA[2]. Table 1 describes the details of statistics of the dataset.

	Train	Dev	Test
Questions	208	25	150
Answers	1701	234	1107

Table 1: Statistics of dataset of QA task.

Evaluation Metrics: For BioNLP 2019 QA task, organizers employ two measurements: accuracy and spearman. The evaluation is reported by official evaluation toolkit[3], and accuracy is the main metric. For each experiment, we report the mean values with corresponding standard deviations over 3 repetitions.

3.2 Experimental Setup

The BioBERT we use includes 12 layers (i.e., Transformer blocks), and the dimension of hidden size is 768. The Transformer we use has 3 blocks, each of which contains 16 heads. For each head, the mapped Q, K, and V dimensions are 48. Thus, the input and output dimension of Transformer is 768. We use Adam (Kingma and Ba, 2015) with β_1 = 0.9, β_2 = 0.999 for optimization. The learning rate is 2e-5. The dropout rate is 0.5. The batch size is set to 4. The **BBERT-T**[4] is developed by

[2] https://chiqa.nlm.nih.gov/

[3] https://github.com/abachaa/MEDIQA2019/tree/master/Eval_Scripts

[4] https://github.com/ThreeTreeStar/Question-Answering

PyTorch[5].We use the BioBERT module (Lee et al., 2019) without modifying. The Transformer is developed by ourselves. Computations are run on a single server computer equipped with a GPU.

3.3 Comparisons with baselines

To verify the effectiveness of our model, we compare **BBERT-T** with five baselines listed as follows.

w/o BioBERT: This variant does not use BioBERT. The processed sentence is embedded with pre-trained word embeddings released by (Moen et al., 2013). Then the embedded sequence input into the Transformer layer.

BBERT-LSTM: This variant replaces two Transformers with two BiLSTMs. Question representation H_q and answer representation H_a are passed to two BiLSTMs, respectively. Note that the two BiLSTMs share parameters.

BBERT-CNN: This variant replaces two Transformers with two CNNs with kernel size of {2, 3, 4}. Question representation H_q and answer representation H_a are passed to two CNNs, respectively. Note that the two CNNs share parameters.

BBERT-T (1 block): This variant uses Transformers with one block, rather than three blocks.

w/o CLS: In classification and ranking layer, we do not concatenate h^{CLS} with the output of max pooling q and a. We directly take $[q, a, |q-a|, q \times a]$ as input to softmax layer. The submitted results are from this model. After submitting the results, we found that **BBERT-T** concatenating h^{CLS} as features achieved higher results than **w/o CLS**.

From the results in Table 2, we can conclude followings. First, compared with **BBERT-T**, **BBERT-LSTM** replaces Transformer with BiLSTM which causes the accuracy to drop by 4.97% and the spearman to drop by 3.50%. **BBERT-CNN** replaces Transformer with CNN which causes the accuracy to drop by 4.45% and the spearman to drop by 3.59%. This indicates that long range dependency information extracted by our model is critical to QA task. After all, most of answers in the corpus of BioNLP 2019 QA task have long sequences and the semantic information may be distributed across long distance.

Model	Accuracy (%)	Spearman (%)	Time (sec)
w/o BioBERT	$51.39_{\pm 0.56}$	$-18.48_{\pm 3.39}$	132.83
BBERT-LSTM	$71.03_{\pm 0.91}$	$6.86_{\pm 4.29}$	399.73
BBERT-CNN	$71.55_{\pm 1.73}$	$6.77_{\pm 4.80}$	290.79
BBERT-T (1 block)	$73.83_{\pm 0.25}$	$4.36_{\pm 9.38}$	310.62
w/o CLS	$73.23_{\pm 1.97}$	$7.80_{\pm 5.50}$	369.88
BBERT-T	$76.0_{\pm 1.30}$	$10.36_{\pm 8.80}$	373.43
BBERT-T*	$76.24_{\pm 1.31}$	$17.12_{\pm 9.66}$	373.43

Table 2: Comparisons with baselines, * stands for using ensemble by averaging the last 4 epoch output probabilities. ± denotes standard deviation,

Second, compared with **BBERT-T**, **w/o BioBERT** decreases the accuracy by 24.61% and the spearman by 28.84%. This indicates that medical domain information is important for BioNLP 2019 QA task.

Third, comparing **BBERT-T** with **w/o CLS**, we can see that without the h^{CLS} feature decreases the accuracy and spearman. In BioBERT, h^{CLS} is originally used as features to classify whether given two sentences is adjacent. Similarly, in our model, h^{CLS} contains important information about the relationship between question and answer, which is critical features to QA task.

Finally, compared with **BBERT-T (1 block)**, **BBERT-T** has a higher complexity but achieves a better accuracy, which illustrates that the structure of three blocks is necessary.

3.4 Effects of architecture

To better understand the architecture of **BBERT-T**, we compare it with three variants:

w/o share: This variant uses two separate Transformers with different parameters .

BBERT-T (att): This variant replaces the max pooling with an attention mechanism. Take the question as example, we calculate the attention weight γ_i for the i th position in the output of Transformer E_q as follows:

$$\gamma_i = \text{soft max}(\tanh(W_\gamma E_i^q + b_\gamma)) \quad (11)$$

where $W_\gamma \in \mathbb{R}^d$ and b_γ are trainable parameters. Then the question features q is defined as follows:

[5] https://pytorch.org/

$$q = \sum_{i=1}^{i=l_q} \gamma_i h_i^q \quad (12)$$

In the same way, the answer feature a is achieved.

BBERT-T (mean): This variant replaces the max pooling with a mean pooling.

From the results in Table 3, we can see that not sharing parameters between the two Transformers might lose connection between question and answer, leading to performance decrease. Using attention mechanism to generate question and answer features achieves a worse results than using max pooling. The reason might be that the self-attention structures of Transformer make each position of output equally important. Therefore, attention mechanism cannot learn the effective weight for each position. This can be further verified by similar accuracy of the mean pooling that gives equal weight of each position.

Model	Accuracy (%)	Spearman (%)
w/o share	$73.83_{\pm 0.32}$	$12.90_{\pm 3.29}$
BBERT-T (att)	$72.99_{\pm 0.65}$	$16.55_{\pm 5.08}$
BBERT-T (mean)	$73.62_{\pm 0.56}$	$9.96_{\pm 8.10}$
BBERT-T	$76.0_{\pm 1.30}$	$10.36_{\pm 8.80}$

Table 3: Effects of architecture.

3.5 Effects of pre-training corpus knowledge

To explore the effects of large-scale pre-training corpus knowledge, we compare our BBERT-T with its two variants:

w/o Bio: This variant replaces BioBERT with BERT.

Model	Accuracy (%)	Spearman (%)
w/o BioBERT	$51.39_{\pm 0.56}$	$-18.48_{\pm 3.39}$
w/o Bio	$71.09_{\pm 0.48}$	$18.89_{\pm 8.10}$
BBERT-T	$76.0_{\pm 1.30}$	$10.36_{\pm 8.80}$

Table 4: Effects of large-scale pre-training corpus knowledge on performance on the QA dataset.

From Table 4, comparing **w/o Bio** with **w/o BioBERT**, we can conclude that the contextualized representations generating by BERT do provide the semantic information between question and answer. BioBERT, having same model structure as BERT, is pre-trained on large scale medical corpus, which could generate medical domain-specific representations. For BioNLP 2019 QA task in medical domain, applying medical domain-specific representations is more effective than open domain representations.

3.6 Effect of reference loss

To investigate the effects of weighted cross entropy loss, we use the cross-entropy loss to train our model and the results are shown in Table 5.

Model	Accuracy (%)	Spearman (%)
Cross-entropy	$73.56_{\pm 0.70}$	$7.76_{\pm 4.86}$
BBERT-T	$76.0_{\pm 1.30}$	$10.36_{\pm 8.80}$

Table 5: Effects of large-scale pre-training corpus knowledge on performance on the QA dataset.

From Table 5, we can observe that cross-entropy gets worse results than weighted cross-entropy, which illustrates that weighted cross-entropy could take advantage of the reference score during training. The candidate answers with higher reference scores are more relevant. Weighted cross-entropy assigns a higher weight to the loss of the correct answer with higher reference score and loss of the incorrect answer with lower reference score, which makes the model more robust.

4 Conclusion

We propose BioBERT Transformer model which applies two Transformers to catch the association between question and answer. Experimental results show that our model benefits from the long range dependency information between words in question and answer and that medical domain-specific contextualized representations generated by BioBERT can effectively improve the performance of QA task. We evaluate on BioNLP 2019 QA test dataset with official evaluation toolkit. And our proposed method achieves the accuracy of 76.24% and spearman of 17.12%.

Acknowledgments

This work is supported by National Natural Science Foundation of China (No. 61772109) and the Humanities and Social Science Fund of Ministry of Education of China (No. 17YJA740076).

References

Asma Ben Abacha, Eugene Agichtein, Yuval Pinter and Dina Demner-Fushman1. 2017. Overview of the Medical Question Answering Task at TREC 2017 LiveQA. *text retrieval conference.*

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *In Advances in Neural Information Processing Systems,* pages 6000–6010.

Asma Ben Abacha, Chaitanya Shivade, and Dina Demner-Fushman. 2019. Overview of the MEDIQA 2019 Shared Task on Textual Inference, Question Entailment and Question Answering. ACL-BioNLP 2019.

Di Wang, and Eric Nyberg. 2017. Cmu oaqa at trec 2017 liveqa: A neural dual entailment approach for question paraphrase identi_cation. *In Proceedings of The Twenty-Sixth Text REtrieval Conference*, TREC 2017, Gaithersburg, Maryland, USA.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805.*

Jimmy Lei Ba, Jamie Ryan Kiros and Geoffrey E. Hinton. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450.*

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2019. Biobert: pre-trained biomedical language representation model for biomedicaltext mining. *arXiv preprint arXiv: 1901.08746.*

Kaiming He, Xiangyu Zhang, Shaoqing Ren and Jian Sun. 2016. Deep Residual Learning for Image Recognition. *computer vision and pattern recognition*,770-778.

Kateryna Tymoshenko and Alessandro Moschitti. 2015 Assessing the Impact of Syntactic and Semantic Structures for Answer Passages Reranking. *conference on information and knowledge management:* 1451-1460.

Kingma, Diederik P., and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980.*

Martin Fajcik, Lukáš, Burget, and Pavel Smrz. 2019. BUT-FIT at SemEval-2019 Task 7: Determining the Rumour Stance with Pre-Trained Deep Bidirectional Transformers. *arXiv preprint arXiv:1902.10126.*

Matthew E Peters, Waleed Ammar, Chandra Bhagavatula, and Russell Power. 2017. Semi-supervised sequence tagging with bidirectional language models. *arXiv preprint arXiv:1705.00108.*

Mengqiu Wang, Noah A. Smith and Teruko Mitamura. 2007. What is the Jeopardy Model? A Quasi-Synchronous Grammar for QA. *empirical methods in natural language processing*, 2007: 22-32.

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

Peng Gao, Zhengkai Jiang, Haoxuan You, Pan Lu, Steven Hoi, Xiaogang Wang and Hongsheng Li. 2018. Dynamic Fusion with Intra- and Inter-Modality Attention Flow for Visual Question Answering. *arXiv: Computer Vision and Pattern Recognition.*

Peter Turney. 2000. Types of cost in inductive concept learning. *In Proceedings of the Cost-sensitive Learning Workshop at the 17th International Conference on Machine Learning*, Stanford, CA.

Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage Re-ranking with BERT. *arXiv preprint arXiv:1901.04085 (2019).*

Sampo Pyysalo, Filip Ginter, Hans Moen, Tapio Salakoski, and Sophia Ananiadou. 2013. Distributional semantics resources for biomedical text processing.*Proceedings of LBM.*

Sepp Hochreiter and Jurgen Schmidhuber. 1997. Longshort-term memory. *Neural Computation,* 9:1735–1780.

Yuan Yang, Jingcheng Yu, Ye Hu, Xiaoyao Xu, and Eric Nyberg. 2017. Cmu livemedqa at trec 2017 liveqa: A consumer health question answering system. *In Proceedings of The Twenty-Sixth Text REtrieval Conference, TREC 2017*, Gaithersburg, Maryland, USA.