Coarse-to-Fine Document Ranking for Multi-Document Reading Comprehension with Answer-Completion

Hongyu Liu
School of Computer Science and Technology
Beijing Institute of Technology
Beijing, China
liuhongyu@bit.edu.cn

Shumin Shi ✉
School of Computer Science and Technology
Beijing Institute of Technology
Beijing, China
bjssm@bit.edu.cn

Heyan Huang
School of Computer Science and Technology
Beijing Institute of Technology
Beijing, China
hhy63@bit.edu.cn

***Abstract*—Multi-document machine reading comprehension (MRC) has two characteristics compared with traditional MRC: 1) many documents are irrelevant to the question; 2) the length of the answer is relatively longer. However, in existing models, not only key ranking metrics at different granularity are ignored, but also few current methods can predict the complete answer as they mainly deal with the start and end token of each answer equally. To address these issues, we propose a model that can fuse coarse-to-fine ranking processes based on document chunks to distinguish various documents more effectively. Furthermore, we incorporate an answer-completion strategy to predict complete answers by modifying loss function. The experimental results show that our model for multi-document MRC makes a significant improvement with 7.4% and 13% respectively on Rouge-L and BLEU-4 score, in contrast with the current models on a public Chinese dataset, DuReader.**

***Keywords*-multi-document reading comprehension; document ranking; answer prediction;**

I. Introduction

Machine Reading Comprehension (MRC) is one of the most important natural language processing tasks, which tests the ability of machines to read and understand a document and then answer some questions about the given document. The release of the Stanford Question Answering Dataset (SQuAD) [1] is one of the first large reading comprehension dataset which requires machine to select a text span in the document as an answer to the question. The SQuAD has attracted many researchers in NLP field and has facilitated more research on RNN-based and attention mechanism models and more complicated datasets.

The traditional MRC architecture requires customized question and document modeling layers to capture the interaction between questions and given documents. In recent years, various attention mechanisms have been proposed to obtain better question-awareness document representation [2], [3]. Since last year, Bidirectional Encoder Representations from Transformers (BERT) [4] introduces a powerful text representation architecture that performs very well on multiple NLP tasks, especially in MRC task.

On the other hand, the creation of larger and more complex datasets also contributes to the development of MRC. Compared to SQuAD, the recently proposed datasets [5], [6] are based on more realistic scenarios, such as multi-document reading comprehension. In the meantime, Chinese reading comprehension has attracted more attention.

In this paper, We demonstrate the validity of the proposed model on a Chinese multi-document reading comprehension dataset, DuReader [5]. The main contributions can be summarized as follow:

1. We introduce a coarse-to-fine document ranking approach to rank documents with different granularity.

2. We propose an answer-completion strategy to empower model to focus on predicting more complete answer.

3. We conduct sufficient experiments on DuReader. The results show that our model achieves great performance on multi-document MRC task.

The remainder of this paper is organized as follows: The second section introduces the latest progress of machine reading comprehension. Detailed of our model will be given in the third section. In the fourth section, we compare the experiments results of our model with other baselines to prove the validity of our proposed model. In the final section, we summarize our work and propose future research directions.

II. Related Work

Since the release of the SQuAD, many researchers have proposed a variety of models. Seo et al. [2] first introduced the idea of bidirectional attention which can compute question-awareness document representation and document-awareness question representation simultaneously for detecting the soft alignment between questions and documents. Wang et al. [3] proposed multi-granularity hierarchical attention and the model surpassed human performance on SQuAD.

In addition to MRC task, attention mechanism plays an important role in many tasks. Google proposed BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding [4] in October 2018 which achieved the state-of-the-art for eleven NLP tasks including MRC tasks.

Compared with SQuAD, real-world scenarios have multiple documents for each question so that it requires the machine to have the ability to filter irrelevant documents and aggregate all relevant document information to get the final answer. As a result, there are many multi-document

978-1-7281-5014-7/19/$31.00 ©2019 IEEE

datasets have been released [5], [6]. Some researchers views document ranking as an effective way to solve multi-document reading comprehension [7], [8]. Previous works rank documents at document or paragraph level and utilize distant supervision to distinguish the positive and negative examples (a document or paragraph containing a text span that exact match with ground truth answer will be considered as a positive sample, otherwise it will be considered as a negative sample). However, the length of the answer is relatively longer so that other documents can not exact match with the ground truth answer except the ground truth document, it is difficult to determine whether a document is a positive or negative example. At the same time, researchers pay more attention to new model architecture which can take multiple documents as input. At present, there are two categories of the current approaches: one is the pipeline-based approaches [9], [10], the other is the multi-task approaches [11], [12]. The pipeline approach firstly gives each document a ranking score and then passes all documents to the MRC model, the final answer is generated by the joint score of the document and the candidate answers. The multi-task approaches can select documents and answers simultaneously which the model can capture richer semantic information from the multi-task shared encoding layers.

There have been some works on Chinese multi-document machine reading comprehension. Wang et al. [12] focused on answer prediction and proposed to model answer content and conduct cross-passage answer verification. Yan et al. [13] introduced a deep cascade model to balance effectiveness and efficiency in real-world scenarios. Liu et al. [14] integrated minimum risk training into their model to mitigate the loss deviation caused by the answer appearing in the document multiple times during training phase.

III. Our Model

In this section, we first introduce BERT architecture for reading comprehension that we used as our backbone network. Then we present our model which consider both document-aspect and answer-aspect for multi-document MRS task. Figure 1 shows our model architecture.

A. BERT Architecture for Reading Comprehension

BERT is a multi-layer bidirectional encoder representation from transformers that can be trained on a large corpus with two novel unsupervised pre-training tasks. In this paper, we use BERT as our backbone network.

1) Input Representation: In multi-document reading comprehension scenario, one question $\boldsymbol{q}$ corresponds to a set of documents $\boldsymbol{D} = \{\boldsymbol{d_1}, ..., \boldsymbol{d_{N_D}}\}$. Because BERT limits the maximum input length, we use each document as the input to our model respectively instead of concatenating all documents into a new document. The input is each question $\boldsymbol{q}$ and one corresponding document $\boldsymbol{d_i}$ and finally we get $\boldsymbol{N_D}$ candidate answers for each question.

Although we don't concatenate all documents into one document, some documents are still exceed the input limit.

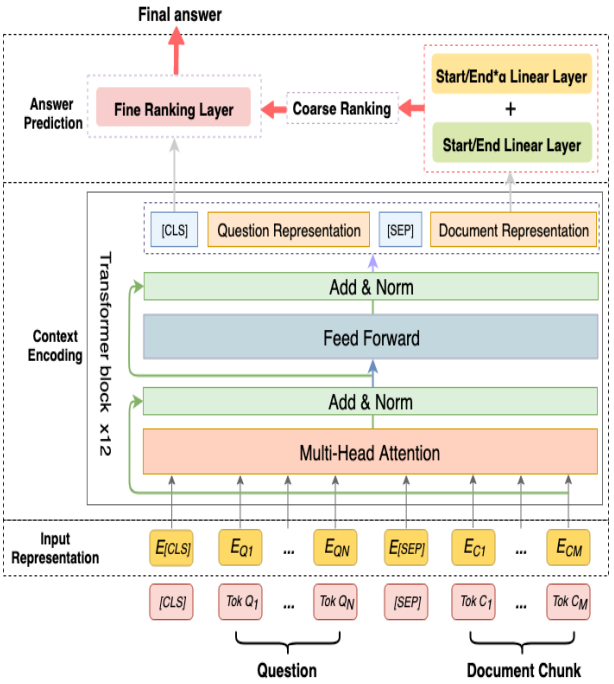

Figure 1. Model architecture with coarse-to-fine ranking and answer-completion

Following [4], we slide a window of length l with a stride s over the document $\boldsymbol{d_i}$ and produce several document chunks $\boldsymbol{C_i} = \{\boldsymbol{c_{i1}}, ..., \boldsymbol{c_{iN_C}}\}$, where $\boldsymbol{N_C} = \lceil \frac{\boldsymbol{L_{di}} - l + s}{s} \rceil$ is the number of document chunks for document $\boldsymbol{d_i}$ and $\boldsymbol{L_{di}}$ is the length of the document.

The input sequence consists of a `[CLS]` token, the tokenized question $\boldsymbol{q}$, a `[SEP]` token, the tokenized document chunk $\boldsymbol{c}$ and a `[SEP]` token, where `[CLS]` is a classification token and `[SEP]` is another token for differentiating sentences. For each token in questions and document chunks, its input representation is constructed by summing the token, segment and position embeddings. We recommend readers to read [4] for more details.

2) Context Encoding: We use 12 layers of transformer blocks to encode question, document chunk and the interaction between the question and document chunk simultaneously. Due to space limitations, we will omit the description of the transformer architecture. The details of the transformer block can be found in the original paper [15]. We denote the document chunk representation in the transformer output layer as $\boldsymbol{C}$.

3) Answer Prediction: There are two simple linear layers followed by the transformer blocks to predict the probability of each token in the context being the start or the end token of an answer span. More specifically, the unnormalized score and probabilities distribution of start and end position are modeled as:

$$\boldsymbol{g^s} = \boldsymbol{W_s C},\ \boldsymbol{p^s} = \boldsymbol{softmax}(\boldsymbol{g^s}) \tag{1}$$

$$\boldsymbol{g^e} = \boldsymbol{W_e C},\ \boldsymbol{p^e} = \boldsymbol{softmax}(\boldsymbol{g^e}) \tag{2}$$

where p^s and p^e are the probabilities for start and end position of answer spans, and W_s and W_e are trainable parameter matrices.

The objective function is defined as the cross-entropy of the predicted probability indexed by true start and end indices, averaged over all the training samples:

$$L_{ANS} = -\frac{1}{N}\sum_{i=1}^{n}\left[\log p^s\left(y^i_{start}\right) + \log p^e\left(y^i_{end}\right)\right] \tag{3}$$

where N is the number of samples in the dataset and y^i_{start},y^i_{end} are the gold start and end positions.

During inference, the probability for a text span from token i to token j being the answer is given by:

$$p(i,j) = softmax(g^s(i) + g^e(j)) \tag{4}$$

For each document corresponding to one question, every chunk from that document will produce a candidate answer with an answer probability. We take the highest probability answer of all candidate answers from all chunks as the final prediction for current document. In the end, we obtain N_D candidate answers for each question and denote $P(A_i)$ as the probability of the answer from the i^{th} document.

B. Document Ranking

Document ranking is a sub-task in multi-document reading comprehension which aims to rank the documents to get better answer from the candidate answers. Previous approaches first use statistical information or shallow semantic information to prune or filter irrelevant documents, and then utilize deep semantic information to rank the remaining documents to distinguish the correlation between documents and questions. In this work, we propose an coarse-to-fine method to directly rank documents using statistical or shallow semantic information and deep semantic information.

1) Coarse Document Ranking: Each question and corresponding documents in DuReader come from the Baidu search engine. Each question corresponds to five documents. The order of the documents in DuReader is the actual order in the search engine. When people use search engines in their daily lives, they will pay more attention to the top search results, because the more advanced search results are more relevant to the question that can provide more valuable information. We believe that search engines can reflect the statistical or shallow semantic information between questions and documents to some extent.

Therefore, we count the frequency at which the answer appears in each document in the training set. The statistical result is shown in Table I. As a result, we set different weights as documents prior knowledge for the answers from different documents to represent the importance of the documents, rather than treating each document equally. We denote the i^{th} document prior probability as dp_i which is derived from the statistics of the training set. We believe it can reflect statistical or shallow semantic information between the question and the document. The probability for a candidate answer from the i^{th} documents is updated by:

Table I
THE FREQUENCY AND PROPORTION OF ANSWERS APPEAR IN DIFFERENT DOCUMENTS

	Frequency	Proportion
Doc1	114941	44.5%
Doc2	60240	23.3%
Doc3	40230	15.6%
Doc4	25060	9.7%
Doc5	18004	6.9%

$$P\left(A_i\right) = \frac{e^{dp_i \cdot P(A_i)}}{\sum_{n=1}^{N_D} e^{dp_n \cdot P(A_n)}} \tag{5}$$

At the same time, we should note that the above search engine order is not always available in other datasets. So we use the normalized token-level $Recall$ value between the question and document title as the prior probability of document to verify the validity of our hypothesis, that is, statistical or shallow semantic information can also be used for ranking document.

2) Fine Document Ranking: Previous works rank documents at document or paragraph level and utilize distant supervision to distinguish the positive and negative examples. Our work is different from them. First, only one document chunk is visible to the model at a time. In order to distinguish semantic information at a finer granularity, we decide to rank documents at chunk-level. Furthermore, the length of the answer is relatively longer (i.e. the average answer length is greater than 100 words) in Chinese multi-document reading comprehension, we can't utilize distant supervision to label the positive samples because there is no document have a text span exact match with the ground truth answer except the ground truth document. Therefore, we propose a novel document ranking method suitable for Chinese multi-document reading comprehension. There are N_D documents for each question. For the document where the ground truth answer is located, the chunk from that document containing the ground truth answer is labeled as a positive sample. And for other documents, the document chunk with the smallest $F1$ value for ground truth answer is labeled as a negative sample. Then the document ranking can be viewed as two-classification task. Suppose that C_{ls} is the hidden representation of the token `[CLS]` from the transformer output, which can be viewed as the deep semantic representation of the question and input document chunk. The probability cp that the input document chunk is labeled as a positive sample is modeled as:

$$cp = softmax(W_c C_{ls}) \tag{6}$$

where W_c is trainable parameter matrix.

we use the cross-entropy loss as the document ranking objective:

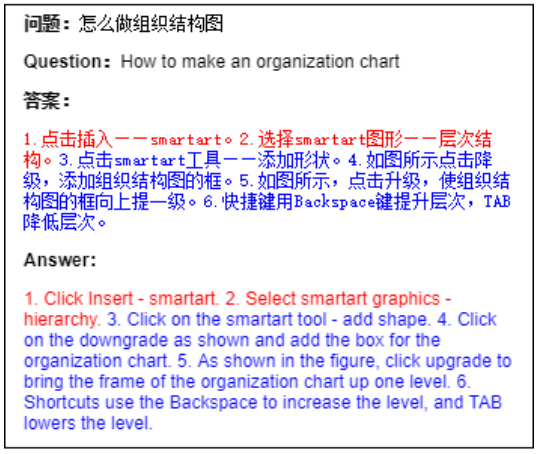

Figure 2. An incomplete answer example in DuReader

$$L_{DOC} = -\frac{1}{K}\sum_{i=1}^{k}\log\left(cp\right) \tag{7}$$

where K is the number of chunks we selected from the dataset.

we denote cp_i as the chunk probability of the document chunk from the i^{th} document. The probability for a candidate answer from the i^{th} documents is further updated by:

$$P\left(A_i\right) = \frac{e^{cp_i \cdot P(A_i)}}{\sum_{n=1}^{N_D} e^{cp_n \cdot P(A_n)}} \tag{8}$$

C. Answer-completion Strategy

1) End token more important: We discover that in cases where the truth answer contains multiple sentences, the model sometimes only predict the first few sentences. Figure 2 shows a qualitative example of this phenomenon. The model only predicts the red answer and discards blue answer. In order to address this problem, we modify the loss function (Eq.3) of the model so that the model can pay more attention to the end token of answers in the training phase. We set the hyperparameter α to indicate the importance of the end token. The modified unnormalized score, probabilities distribution of start and end position as well as the loss function is given by:

$$g^{s\prime} = W_s' C,\ p^{s\prime} = softmax(g^{s\prime}) \tag{9}$$

$$g^{e\prime} = W_e' C,\ p^{e\prime} = softmax(g^{e\prime}) \tag{10}$$

$$L'_{ANS} = -\frac{1}{N}\sum_{i=1}^{n}[\log p^{s\prime}(y^i_{start}) + \log p^{e\prime}(y^i_{end}) \cdot \alpha] \tag{11}$$

It has been experimentally verified that the modified loss function may hurt the performance of our model correctly predicts the start token. So we jointly consider the impact of two loss functions on our model (i.e. we use two different prediction layer), the probability for a text span from token i to token j being the answer (Eq.4) is modified by:

$$p(i,j) = softmax(g^s(i) + g^e(j) + g^{s\prime}(i) + g^{e\prime}(j)) \tag{12}$$

D. Training and Inference

During training, we take a multi-task learning approach [16], sharing the parameters of transformer blocks with a joint objective function defined as:

$$L = L_{DOC} + L_{ANS} + L'_{ANS} \tag{13}$$

All samples are used to train document ranking module, but only the positive samples are passed to subsequent modules for training the prediction layer, the negative samples are discarded.

During inference, we take the coarse ranking probability dpi, fine ranking probability cpi, and answer probability $P(A_i)$ into account. The final answer probability is calculated by Eq.12, Eq.5 and Eq.8. We compare the probability across all chunks from the same instance, and choose the final answer according to the final probability.

IV. Experiment and Analysis

A. Dataset

We evaluate our model on a Chinese multi-document comprehension reading dataset, DuReader[1]. DuReader is the largest Chinese MRC dataset, which contains 200k question, 1M documents and more than 420k human-annotated answers. Each question has 5 evidence documents. All the questions and documents comes from Chinese search engine Baidu.

B. Data Preprocessing

Each document contains several paragraphs. We calculate the maximum $F1$ value of the corresponding question at paragraph level for each document. Then we select the top-N paragraphs that have largest $F1$ value with each question. These paragraphs are reassembled into a new pruned document in the order of the original document. Through this preprocessing, a large number of irrelevant paragraphs can be filtered out.

C. Evaluation Metrics

According to the evaluation metrics of the DuReader, we evaluate our model via character-level BLEU-4 and Rouge-L.

D. Baselines

Because there haven't any published work research on DuReader based on BERT. We report the experimental results on DuReader in the published work and construct BERT baseline[2] ourselves without document ranking and answer-completion strategy.

[1]https://github.com/baidu/DuReader
[2]https://github.com/huggingface/pytorch-transformers

Table II
PERFORMANCE OF OUR MODEL AND COMPETING MODELS ON THE DUREADER

Model	Rouge-L	BLEU-4
BiDAF[2]	39.0	31.8
PR+BiDAF[12]	41.81	37.55
V-NET[12]	44.18	40.97
Deep Cascade[13]	50.71	49.39
MRT[14]	51.09	43.76
BERT	49.61	44.62
Our model	**53.27**	**50.42**
Our model*	**53.13**	**54.63**
Human performance	57.4	56.1

E. Implementation Details

We use BERT as our baseline. We set up several experiments with the default BERT models configurations. And finally we set hyper-parameters for our proposed model which have the best Rouge-L performance. We use Adam optimizer with a learning rate of 3e-5 and warmup over the first 10% steps to train for 2 epochs. The batch size is 6, the document chunk length is 512 and the document stride length is 128. The document prior probability is 0.45,0.23,0.16,0.09,0.07 for each document, the values are from the statistical information of the training set. The end token importance hyperparameter $\boldsymbol{\alpha}$ is 2.

F. Main Results

The results of our model on DuReader are summarized in Table II.

Because the DuReader *test set 1.0* is temporarily closed, we show the performance of our model on DuReader development set and the performance of other published models are tested on DuReader *test set 1.0*. According to the previously published data, the test set and development set have similar performance. As we can see in Table II, BERT baseline achieves comparable performance to the currently published models. The coarse-to-fine document ranking and answering strategy we proposed on our model further improve the performance, which improve 7.4% in Rougel-L and 13% on BLEU-4. The last row in Table II is the performance of our model on DuReader *test set 2.0*. DuReader *test set 2.0* consists of the difficult samples for current models in *test set 1.0*, which is more challenging than *1.0*. Our model also achieves great performance on *test set 2.0* which can demonstrate the validity of our proposed model. We will analyze our model in detail in the next section.

G. Ablation Study

To further study the effectiveness of our model, we conduct an in-depth ablation study on the development set of DuReader, which is shown in Table III.

We first evaluate the answer-completion strategy by ablating the modified loss function so that the modified loss will not be used during training and testing. Then we remove the fine and coarse ranking in order to test the necessity of the coarse-to-fine ranking.

Table III
ABLATION STUDY ON DUREADER DEVELOPMENT SET

Model	Rouge-L	△	BLEU-4	△
Our model	**53.27**	-	**50.42**	-
- answering strategy	52.56	-0.71	49.67	-0.75
- fine ranking	52.11	-0.45	49.02	-0.65
- coarse ranking	49.61	-2.5	44.62	-4.4

Table IV
COMPARISON AMONG DIFFERENT DOCUMENTS AND DIFFERENT RANKING APPROACH ON DUREADER DEVELOPMENT SET

	Rouge-L	BLEU-4
Doc1	**50.18**	**48.27**
Doc2	44.4	41.31
Doc3	40.42	35.87
Doc4	36.11	30.2
Doc5	34.02	27.97
Multi-doc	49.61	44.62
Coarse ranking (Re)	50.83	48.33
Coarse ranking (PK)	52.11	49.02
PK + fine ranking	**52.56**	**49.67**

From Table III, we can see that the coarse-to-fine ranking can help the model to distinguish between different documents and focus on the documents related to the question. The coarse ranking can effectively help model filter irrelevant documents and the fine ranking can further improve the performance because it can capture deep semantic relation between questions and documents which maybe ignored in the coarse ranking. We will further analyze the document ranking and answer-completion strategy respectively in the following section.

H. Document Ranking

As is shown in Table IV, we evaluate the performance of the model on different ranking method. In order to verify our hypothesis (i.e. the order of search engines can cover some statistical or shallow semantic information), we evaluate our model on different documents. We can see that the performance of the model using the last document is much lower than using the first document which can indicate the top ranked documents can provide more reliable answers. And equally considering all the documents without any document ranking does not work well because the model can't filter the irrelevant documents very well. After adding the document prior probability as prior knowledge (PK), the performance of the model can be effectively improved. But the prior knowledge is not always available in other datasets. We also experiment with a simple coarse ranking method (Re) that rank documents by using the $\boldsymbol{Recall}$ value between question and the document title to prove the validity of the coarse ranking. We can see that although the performance is a little worse than prior knowledge, because the order of the search engines involves complex ranking mechanisms which can provide us with a good coarse ranking effect. However, it is still better than the performance without ranking. After adding the fine ranking, the model achieves the best results which shows that the coarse ranking and

Table V
THE IMPACT OF THE ANSWER-COMPLETION STRATEGY ON ROUGE-L SCORE ON DUREADER DEVELOPMENT SET

	Zhidao			Search		
	D	E	Y_N	D	E	Y_N
W/O	53.1	58.8	42.6	51.1	51.9	36.6
W/	**53.8**	**59.5**	**44.4**	**51.5**	**53.1**	**38.1**

the fine ranking can capture the correlation between question and documents at different granularity. The former is statistical information or shallow semantic information and even some additional information that search engines can provide, while the latter is mainly deep semantic information.

I. Answer-completion Strategy

In this section, we discuss in depth the effectiveness of our proposed answering strategy on Rouge-L score. We believe that the modified loss function can make the model pay more attention to the end token of the answer during the training phase. DuReader can be divided into two types according to the source of the data, i.e. *Zhidao* and *Search*. The question types for each subset include *Description*, *Entity* and *Yes-No*. We evaluate the model with and without answering strategy on these six subsets. The results are shown in Table V.

In Table V, W/O and W/ indicates whether the model is combined with the answering strategy or not. D, E, Y_N represent different question types. We can see that the performance of the model with answer-completion strategy is better than the model without it on all six subsets. It demonstrate the answering strategy we proposed can further improve the performance of our model. And we can noticed that for both *Zhidao* and *Search*, where the *Yes-No* question type has the largest improvement. Because for the *Yes-No* question, in addition to predict the answer, we need to predict the category of the views based on the predicted answer (i.e. *Yes, No* and *Depend*). We believe that a more complete answer can further improve the performance of the classifier, so the *Yes-No* question type has the largest improvement.

V. CONCLUSION

In this paper, we consider both document-aspect and answer-aspect for multi-document reading comprehension. First of all, we propose a coarse-to-fine document ranking based on document chunks to measure the relevance of questions and documents at different granularity. And then the answer-completion strategy can enable our model to pay more attention to the end token for detecting one entire answer. The experimental results demonstrate that our model achieves great performance in the multi-document MRC task. We will focus on integrating external knowledge for multi-document MRC task in future work.

ACKNOWLEDGMENT

This work is supported by the National Natural Science Foundation of China (No.61671064) and the National Key RD Program of China under Grant (No.2018YFC0831704).

REFERENCES

[1] Rajpurkar, Pranav, et al. Squad: 100,000+ questions for machine comprehension of text. arXiv preprint arXiv:1606.05250, 2016.

[2] Seo, Minjoon, et al. Bidirectional attention flow for machine comprehension. arXiv preprint arXiv:1611.01603, 2016.

[3] Wang, Wei, Ming Yan, and Chen Wu. Multi-granularity hierarchical attention fusion networks for reading comprehension and question answering. arXiv preprint arXiv:1811.11934, 2018.

[4] Devlin, Jacob, et al. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805, 2018.

[5] He, Wei, et al. Dureader: a chinese machine reading comprehension dataset from real-world applications. arXiv preprint arXiv:1711.05073, 2017.

[6] Nguyen, Tri, et al. MS MARCO: A human generated machine reading comprehension dataset. arXiv preprint arXiv:1611.09268, 2016.

[7] Htut, Phu Mon, Samuel R. Bowman, and Kyunghyun Cho. Training a ranking function for open-domain question answering. arXiv preprint arXiv:1804.04264, 2018.

[8] Lee, Jinhyuk, et al. Ranking Paragraphs for Improving Answer Recall in Open-Domain Question Answering. Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. 2018: 565-569.

[9] Chen, Danqi, et al. Reading Wikipedia to Answer Open-Domain Questions. Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 2017: 1870-1879.

[10] Wang, Shuohang, et al. R3: Reinforced Ranker-Reader for Open-Domain Question Answering. Thirty-Second AAAI Conference on Artificial Intelligence. 2018: 5981-5988.

[11] Clark, Christopher, and Matt Gardner. Simple and Effective Multi-Paragraph Reading Comprehension. Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 2018: 845-855.

[12] Wang, Yizhong, et al. Multi-Passage Machine Reading Comprehension with Cross-Passage Answer Verification. Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 2018: 1918-1927.

[13] Yan, Ming, et al. A Deep Cascade Model for Multi-Document Reading Comprehension. Thirty-Third AAAI Conference on Artificial Intelligence. 2019: 7354-7361.

[14] Liu, Jiahua, et al. A Multi-answer Multi-task Framework for Real-world Machine Reading Comprehension. Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. 2018: 2109-2118.

[15] Vaswani, Ashish, et al. Attention is all you need. Advances in neural information processing systems. 2017: 5598-6008.

[16] Caruana, R. Multitask learning: A knowledge-based source of inductive bias. Machine Learning. 1997: 41-75.