Multiple Perspective Answer Reranking for Multi-Passage Reading Comprehension

Mucheng Ren, Heyan Huang, Ran Wei, Hongyu Liu,
Yu Bai, Yang Wang, and Yang Gao

Beijing Institue of Technology, 100081, Beijing
{rdoctmc,weiranbit,liuhongyu12138,Wnwhiteby,wangyangbit1}@gmail.com
{hhy63,gyang}@bit.edu.cn

Abstract. This study focuses on multi-passage Machine Reading Comprehension (MRC) task. Prior work has shown that retriever, reader pipeline model could improve overall performance. However, the pipeline model relies heavily on retriever component since inferior retrieved documents would significantly degrade the performance. In this study, we proposed a new multi-perspective answer reranking technique that considers all documents to verify the confidence of candidate answers; such nuanced technique can carefully distinguish candidate answers to improve performance. Specifically, we rearrange the order of traditional pipeline model and make a posterior answer reranking instead of prior passage reranking. In addition, new proposed pre-trained language model BERT is also introduced here. Experiments with Chinese multi-passage dataset DuReader show that our model achieves competitive performance.

Keywords: Machine Reading Comprehension · Answer Reranking · BERT.

1 Introduction

Question Answering, as a sub-task of Natural Language Processing, has been a long-standing problem. In recent years, Machine Reading Comprehension (MRC), a task that empowers computers to find useful information and response correct answers from giving questions and related documents in natural language, has drawn a considerable amount of attention. In the beginning, MRC task only focused on cloze style test [5, 8], later followed by single document datasets [15, 16] and complicated open domain datasets [17, 18, 12].

Lots of progress have been achieved over these MRC datasets. Particularly, on benchmark single-passage dataset SQuAD [15], various deep neural network models based on Recurrent Neural Network (RNN) and attention mechanism have been proposed [21, 13]. Some work has already surpassed the performance of human annotators, which can be assumed as a big milestone in MRC filed[6]. However, SQuAD dataset already provides a single passage for each question so that answers can be definitely found in the given passage. Moreover, the length of each given paragraph is relatively short so that there exists a huge

gap between this dataset and real-world scenarios since people usually need to find answer from multiple documents or webpages. Even though SQuAD 2.0 [16] which contains unanswerable questions had been built last year, it is still limited by practical difficulties. Therefore, several studies [17, 18, 12] start to build a more realistic MRC dataset: Read multiple related documents to answer one question which is called as multi-passage datasets.

Compared with single-passage datasets, the most critical problem in multi-passage MRC is noisy input data: for each question, all given passages are related but not essential which means every document describes a common topic but in different ways. Therefore, too much related information may confuse the model significantly. In general, multi-passage MRC task is usually done by two categories of approaches: 1) The pipeline approach usually separates whole MRC task into two subtask: passage selection and extractive reading comprehension like SQuAD. Given a question and multiple related documents, the most important document should be chosen by passage reranking techniques, then send it into MRC model to figure out the answer [11, 22], our work follows this approach; 2) Joint learning approach integrates these two subtasks so that they can be trained simultaneously [14, 3, 23, 9].

Be different from the previous pipeline method, our pipeline did not follow the traditional processing order, we discard passage selection component. Instead, we firstly do answer prediction for each passage to get a set of answer candidates, then an answer reranking component will be applied to determine confidence for each predicted answer and answer with the highest confidence is the final output answer. This multi-perspective technique allows meticulous sorting for candidate answers so that overall performance can be improved. In addition, we abandon traditional neural network entirely and choose Bidirectional Encoder Representations from Transformers (BERT) [6] pretrained language model as basic computational units. BERT is a newly proposed pretrained language model, it consists of numerous transformers [2] whose working principle is multi-head attention which can ensure each word can be greatly represented according to its context. Since it is a pretrained language model, it can be simply adopted into different NLP tasks by finetuning it, so far BERT already monopolized almost every MRC test datasets [15–17].

Our contribution is two-fold:

1. Firstly, we designed a novel pipeline model in a reversed order and proposed a multi-perspective answer reranking technique to verify the confidence of answer candidates. With confidence verification, superior answers can be explicitly distinguished with inferior ones.

2. Secondly, we explored the possibility that is applying pretrained language model BERT into multi-passage MRC task. More importantly, we chose to adopt different BERTs in whole pipeline and demonstrated the effectiveness of pretrained language model.

We conduct extensive experiments on DuReader [18] dataset. The results show that our BERT based pipeline model outperforms the baseline models by a large margin and confidence verification works well. Our project code is available[1].

2 Related work

Multi-passage MRC
In recent year, multi-passage MRC research has drawn great attention [17, 18, 12, 14, 11, 22, 3, 23, 9]. Be different from single passage datasets, multi-passage MRC needs model becomes more robust to noisy data. The most straightforward approach is to concatenate all passages and find the answer from the integrated one [24]. Generally, there are two categories of approaches explored in multi-passage MRC: pipeline model and joint training model. For pipeline model, most models firstly filter out the most relevant passage by using a TF-IDF based ranker or a neural network based ranker, then pass it into a neural reader [22, 4, 10, 20]. However, the performance of the pipeline approach suffers from the document ranking model, since posterior reading comprehension component can not extract correct answer if filtered documents are incorrect. For joint learning approach [14, 3, 23, 9], it considers all the passages and selects the best answer by comparing confidence scores. Wang (2018) [23] propose a cross-passage answer verification for more accurate prediction. Tan (2018) [3] propose an extraction-then-synthesis framework to synthesize answers from extraction result. Y.Ming (2018) [14] further consider a proper trade-off between the pipeline method and joint learning method, it uses cascade learning to eliminate useless passages in advance and identify the best answer on remaining passage. Our model follows the pipeline model approach which trains each component separately, however we consider all passages like joint learning method does and propose a new answer confidence verification method.
Pretrained LM
Devlin (2018) [6] propose Bidirectional Encoder Representations from Transformers (BERT), a new language representation model that obtains state-of-the-art results on eleven natural language processing tasks. Nowadays, BERT has been widely adopted in various fields. For example, Hu [20] proposed a RE3QA that adopted BERT into Retrieval and Reader components for multi-document MRC, Yang [19] used BERT as new document reader in open domain question answering. Most papers have demonstrated BERT based model with simple fine-tune modification can significantly surpass the performance of traditional neural models in different fields.

3 Proposed Model

The overall pipeline of our model can be seen in Figure 1. In the DuReader dataset, each question is given with several documents, we firstly preprocess

[1] https://github.com/trib-plan/TriB-QA

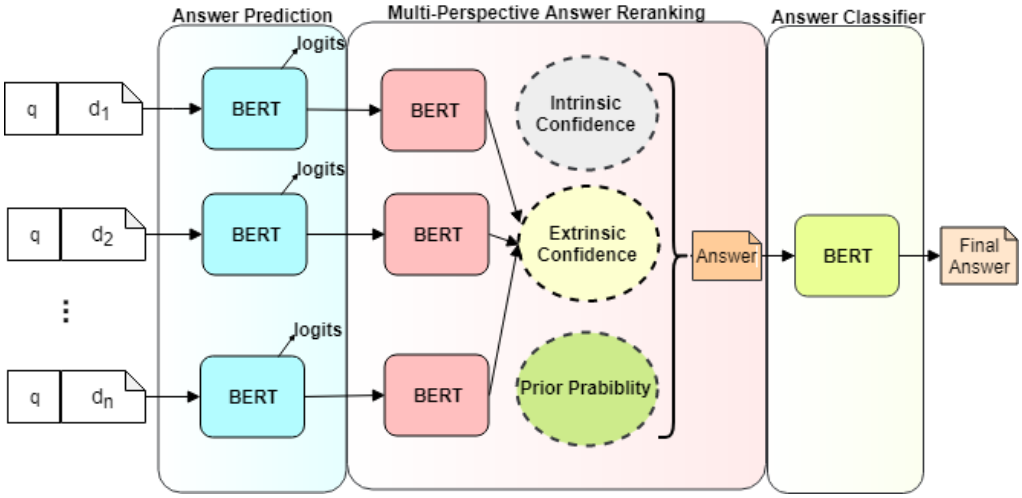

Fig. 1. Overall architecture. Given documents will be concatenated with query which are then fed into the model. Each document will generate an answer candidate. Then multi-perspective answer ranking will verify answer confidence by considering three parts: Intrinsic confidence, Extrinsic confidence and Prior document probability. The answer candidate with the highest confidence will be selected as the answer and another answer classifier will be applied if question type is Yes/No. The whole pipeline model is trained independently.

the dataset by using several statistical tricks, then feed each question and several question-related passages into Answer Prediction module to get answers for each passage, then generated answers and query will be further sent into Answer Reranking module to compare the confidence scores between each answer, and answer with highest confidence will be selected as final output answer.In addition,if the query type is Yes/No type, another Yes/No/Depends Classifier module will be applied at the end to determine the polarity of the output answer.

Particularly, during the training process, each module is trained separately. In the answer prediction part, we firstly preprocess the given documents for one question to ensure the input documents must contain the correct given answer span, therefore this part can be trained like SQuAD dataset. In the answer reranking module, we design a multi-perspective ranking technique and use self-made labels to rank generate answers.At the last, in the Yes/No/Depend classifier module, only yes/no type questions in datasets would be selected and given labels would be used here for training purpose.

3.1 Answer Prediction

Given a query $\boldsymbol{q}$ and multiple corresponding pre-processed documents $\boldsymbol{C} = \{\boldsymbol{c_1}, ..., \boldsymbol{c_n}\}$ where n is 5 at most, the Answer Prediction component aims to

generate one set of answer candidates $\boldsymbol{A} = \{\boldsymbol{a_1}, ..., \boldsymbol{a_n}\}$. This can be achieved by following procedures.Firstly, we encode query with every document together by using pretrained Transformer blocks [6]. Particularly we concatenate query and document as $\{[CLS]; \mathbf{q}; [SEP]; \mathbf{c}; [SEP]\}$ where $[CLS]$ is a token for classification token and $[SEP]$ is another token for separating different sentences.

Next, the final hidden states from BERT for the $\boldsymbol{i_{th}}$ input token can be denoted as $\boldsymbol{T_i}$.In order to predict answer span with highest probability, we calculated the probability of whether $\boldsymbol{i_{th}}$ input token is start token or end token separately.Particularly, the probability of word i being the start of the answer span can be computed as a dot product between T_i and S followed by a SoftMax layer,where S is the learnable matrix that we should train. Similarly, the probability of word i being the end of the answer span can be calculated by training matrix T. At last, the answer span from word i and word j with highest probability will be selected as final answer.

$$T_i = BERT(q, c) \tag{1}$$

$$P_{Si} = softmax(start\_logits(i)) = \frac{e^{S \times T_i}}{\sum_j e^{S \times T_j}} \tag{2}$$

$$P_{Ei} = softmax(end\_logits(i)) = \frac{e^{E \times T_i}}{\sum_j e^{E \times T_j}} \tag{3}$$

Finally, the training objective for answer prediction component is the loglikelihood of the given answer span labels.

$$L_{AP} = -log(P_S) - log(P_E) \tag{4}$$

3.2 Multi-Perspective Answer Reranking

After the answer candidates $\boldsymbol{A} = \{\boldsymbol{a_1}, ..., \boldsymbol{a_n}\}$ for query $\boldsymbol{q}$ are generated in the answer prediction part, we then input $\boldsymbol{q}$ and $\boldsymbol{A}$ into the multi-perspective answer reranking module. In this part, we combine the intrinsic confidence($\boldsymbol{IC}$), extrinsic confidence($\boldsymbol{EC}$) and statistical distribution for documents($\boldsymbol{\alpha}$) to calculate the final answer confidence.

Statistical distribution for documents The first perspective is the statistical distribution of documents. Since the Dureader dataset is constructed based on real application scenario, all questions are real questions raised by users in Baidu search engine and documents are the results retrieved from it. Therefore, the documents for one query is already sorted in order and it can be argued that the documents that are retrieved by search engine in higher order tend to have better context similarity between query and context, better user acceptance and entity matching. Therefore, we make a statistical analysis of preprocessed dataset and explore the order of documents that contains correct answers. In this way, we define a list of prior probability $\boldsymbol{\alpha} = \{\boldsymbol{\alpha_1}, ..., \boldsymbol{\alpha_n}\}$ in descending order for multiple documents corresponding to one query.

Intrinsic Confidence The second perspective is to determine the quality of generated answer. As mentioned in section 3.1, answer prediction module would transform i_{th} token hidden state into start/end logits. Then token with higher logits will be selected start/end word for an answer span. Be inspired by this, we can assume that the magnitude of logits can represent the intrinsic confidence of generated answer spans in a way. Therefore, we directly sum the start logit and end logit for an answer as intrinsic confidence.

$$ICa_i = start\_logits(a_i) + end\_logits(a_i) \tag{5}$$

Extrinsic Confidence The third perspective aims to analyse the confidence difference between answers generated from different documents. Dureader dataset provides a series of labels that tell us whether this document contains the reference answer. However, such labels are usually misleading because for most cases, document which is labelled as false still contain reliable answers. Therefore, we decide to build a classifier to determine whether the document is trustable or not.

In order to achieve this, we use BERT cascaded with one fine-tune linear layer as classifier, we extract 60K questions (around 200K documents) from train datasets and build prediction labels by following the rules: if the generated answer has good ROUGE-L values compared with given answer (top 30% among all examples), we label the document as 1, otherwise 0 (last 30%). Such self-made labels could ease the misleading label issue. In this way, the extrinsic confidence for a given answer would be final values on $[CLS]$ token.

$$EC_{a_i} = Linear(BERT(q, a_i)) \tag{6}$$

After observing answer from three perspectives, final answer confidence $\boldsymbol{C}$ can be represented as follow.

$$C_{a_i} = ECa_i * softmax(IC_{a_i} * \alpha_i) \tag{7}$$

3.3 Yes or No Discrimination

For sentence classification, a BERT classifier model is applied. The model takes the representation of the answer's $[CLS]$ in and outputs its polarity.

$$P_{a_i} = Linear(BERT(a_i)) \tag{8}$$

Furthermore, we employ an advanced method which binds the answer and the corresponding question together. With the adding question information, our model can makes its decision more wisely. Meanwhile, we change the pretrained model to ERNIE[25] to achieve better performance.

$$P^{'}_{a_i} = Linear(ERNIE(q, a_i)) \tag{9}$$

We then fine-tune the pretrained model to make it suitable for our task. The whole model is trained to minimize the cross-entropy loss.

4 Experiment Setup

In this section, we introduce the setup of our experiment which includes datasets, model settings, data preprocessing and evaluation metrics in detail.

4.1 Datasets

We experiment our model on Chinese multi-passage dataset Dureader [18]. Statistics for official dataset can be found in table 1 and 2 . Particularly, test 1 and test 2 datasets contain mixed data so that the real numbers of evaluated questions are 3398 and 6561 respectively.

4.2 Model settings

We initialize our model using publicly available pytorch version of BERT in Chinese [2]. For simplicity, we adopted same parameters described in [6] except:

For answer prediction component, we set *doc_stride* as 350 ,*max_seq_length* and *max_answer_length* as 512, *batch_size* as 20, *epoch_number* as 2. For answer reranking component, we set *batch_size* as 10, *epoch_number* as 2 or 3,*max_seq_length* as 400. For answer classifier component, we set *batch_size* as 32, *epoch_number* as 4.

We trained our answer prediction model on one NVIDIA RTX 2080Ti GPU and train rest components on one Titan XP GPU.

4.3 Data Preprocessing

The answers for every question in the dataset are summarized by the annotators. Because the current dominant models are extractive, i.e. the answer is a text span from the documents, the dataset provides us with fake answers and the corresponding text spans which have the largest F1 score with true answers for training. First we filtered out the samples where the answer is punctuation or the largest F1 score is less than 0.5. Then we calculated the ROUGE-L score between fake answers and true answers and filtered out the samples that ROUGE-L are less than 50. After sample pruning, we calculated the F1 scores of the question at paragraph-level for each document, and rearrange the top-N paragraphs into a new pruned document in the order of the original document. Finally, the pruned documents are passed to the model for training and testing.

4.4 Evaluation Metrics

In terms of answer evaluation, we adopted the ROUGE and BLEU automatic evaluation method in [1]. The method is improved for machine reading comprehension task. The Evaluation score bases on the score ROUGE-L and BLEU-4. The automatic evaluation method is improved for questions inquiring yes-no

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

Table 1. Dureader Data Distribution

	Train	**Dev**	**Test1**	**Test2**
Zhidao	135366	1060	30000	30000
Search	136208	1179	30000	30000
Total	271574	2239	60000	60000

Table 2. Question Type Distribution

	Fact	**Opinion**	**Total**
Entity	14.4%	13.8%	28.2%
Description	42.8%	21.0%	63.8%
YesNo	2.9%	5.1%	8.0%
Total	60.1%	39.9%	100%

Table 3. Performance of our model and competing models on the DuReader

Model	ROUGE-L	BLEU-4
BiDAF[13]	39.00	31.80
Match-LSTM[21]	39.20	31.90
PR+BiDAF[23]	41.81	37.55
V-NET[23]	44.18	40.97
R-NET[24]	47.71	44.88
Deep Cascade[14]	50.71	49.39
MRT[7]	51.09	43.76
Our model	**55.51**	**55.71**
Human performance	68.68	69.60

opinions and entity lists. If the model correctly answers the yes-no type question or correctly matches entity lists it will receive a score bonus. To some extent, it makes up for the deficiency of traditional ROUGE-L and BLEU-4.To better correlate n-gram overlap with the human judgment for answers to these two question types.

5 Results and Analysis

5.1 Overall Results

Table 3 summarize all the results on the test set of Dureader dataset. It is worth noting that our model and all other models train on the same train set, but test on different test sets due to competition rules[3]. More specifically, the test dataset evaluated by our model is more complicated than others, because this year's test dataset only contains questions that were answered incorrectly last year. As we can see, our best model achieves 55.51 Rouge-L and 55.71 Bleu-4, clearly outperforming previous methods.

5.2 Model Analysis

In this section, we describe **1)** the detailed procedure to achieve final result; **2)** the insufficiency exists in our model.

[3] http://lic2019.ccf.org.cn/read

Table 4. The building procedures for our model

Model	ROUGE-L	BLEU-4
Original Pipeline	**37.15**	**23.41**
+ Data Preprocess	44.54	27.60
+ Increase Answer Length	48.13	46.70
+ Prior Probability α	50.46	52.37
+ Extrinsic Confidence	52.50	54.30
+ Intrinsic Confidence	54.12	55.82
+ Human Constraints	55.30	56.09
Final Results	**55.51**	**55.71**

Building procedure of model Table 4 states detailed building procedure for our model. For the first try, we use traditional pipeline method: select most related document and send it directly into answer prediction model to get answer. Clearly, the result demonstrates traditional pipeline model relies heavily on the quality of document. Therefore, we decide to improve the quality of documents by adopting data pre-processing described in section 4.3. It can be seen that the ROUGE-L value is increased by 7.4 points which verifies the above mentioned quality-matter concept.Meanwhile, we observe that the length of our generated answers is relatively short compared with reference answer so that the BLEU-4 score is far behind ROUGE-L. Hence, we find answers in a larger span and increase the answer length.

Next, we start considering all documents since every document corresponding to one query in the dataset is similar with each other which suggests that these documents should not be filtered out at the beginning. Thus, we adopt prior document probability to verify the answer confidence. The result demonstrates our thought is correct. In order to improve performance of answer reranking, we introduce two more variables: intrinsic confidence and extrinsic confidence. As we can see, it can significantly improve overall performance, suggesting our proposed multi-perspective answer reranking technique is necessary for our model.

At last, in order to make results more competitive, we add some human constraints over the model, including hyperparameter finetuning, punctuation replacement, substitute BERT with ERNIE etc.

Insufficiency exist in model There exist two main critical problems in our model:

- **Answer Length:** Even though we already increase the answer length, some generated answers are still short. This is because BERT can only accept maximum 512 tokens-long documents so that answer span will not exceed this threshold. However, in multi-passage datasets, many documents are much longer than 512 tokens and answers are hundreds of tokens as well. This could be solved by replacing BERT with other neural network like RNN,CNN etc., because the latter one can accept much longer sequence length without consuming too much training sources.

Question	初学波比跳每天多少组	Prior probability	Intrinsic confidence	Extrinsic confidence	Final confidence
Doc_1 Answer	Burpee(波比)是一种高强度,短时间燃烧脂肪,令人心跳率飙升的自重阻力训练动作之一。Burpee结合了深蹲(Squat)、伏地挺身(Push-Ups)及跳跃(Jump)一连串的动作,在短时间内会将心跳率拉升到将近人体最大值。	0.503	-0.36696	0.11551	0.00652
Doc_2 Answer	在你能力范围内可做3~4组波比跳，每组8~20个。	0.2314	8.16496	0.78477	**0.60819**
Doc_3 Answer	波比运动每天做多少个。	0.1414	-3.48146	0.02304	5.47419e-14
Doc_4 Answer	在40秒里,做尽量多的波比,休息20秒,为一组;做20个波比,休息30秒,为一组;不休息,一直做波比,直到力竭为止,为一组;不休息,做多个波比(数量根据自己体能调整),为一组。具体的循环数量,也可以依据自己的训练经验和体能进行调整。	0.1031	8.80785	0.88716	**0.25771**
Doc_5 Answer	一般一天做90个,15个一组,6组,每组间隔30秒。	0.0411	8.93052	0.75435	0.12756
Reference Answers	• 在能力范围内可做3~4组波比跳，每组8~20个，跳多少组，每组多少个看体力。 • 在40秒里，做尽量多的波比，休息20秒，为一组;做20个波比，休息30秒，为一组;不休息，一直做波比，直到力竭为止，为一组;不休息，做多个波比(数量根据自己体能调整)，为一组。				

Fig. 2. A sampled case from Dureader dev set. Our answer prediction module selects answers from each passage. The multi-perspective answer reranking module calculates confidence scores, correct answers with high confidence scores would be selected.

- **Discontinuous answer:** Our model current can only output continuous answers which are directly extracted from documents. However, some question requires abstractive answers or discontinuous answers. Such answers require model to jump around looking for them, i.e. finding the keywords in the documents. This issue can probably solved by teaching mode to focus on essential terms.

5.3 Case Study

To demonstrate powerfulness of our model,we conduct a case study sampled from our model on Dureader development set. For a given query, we present predicted answer candidate for each document with its prior probability, intrinsic confidence and extrinsic confidence. As can be seen in Figure 2, we can make two conclusions.

Firstly, it can be argued that considering all documents is necessary. As can be seen from the figure, either doc2 or doc4 can give us a reasonable answer and doc1 which is labelled as best document does not contain answer. Therefore, if we implement document reranking at first like traditional pipeline model, it has large probability to choose a document that does not contain any answer and other document that may contains answers will be ignored. If we consider all documents, this issue can be avoided.

Secondly, multi-perspective answer reranking technique works as expected. If we only consider prior document probability, doc1 answer will be selected which is wrong. However, after considering intrinsic confidence,we can see that module could identify the quality of answer correctly. Particularly, doc1 and doc3 answers which are wrong answers are given negative scores. Then during

extrinsic confidence verification step, our model can compare confidence between all answer candidate so that superior answers (doc2,doc4,doc5) has larger values in contrast to inferior answers(doc1,doc3). After combining prior probability, intrinsic confidence and extrinsic confidence, the answer with highest final confidence can be correctly selected. Therefore, it can be concluded that these three perspectives are compensated with each other, such multi-perspective answer reranking technique can indeed improve overall performance.

6 Conclusion

In this study, we proposed a new multi-perspective answer reranking technique. Our refined pipeline model can verify the confidence of candidate answers such that superior answers can be distinctly distinguished with inferior answers. Specifically, we make a posterior answer reranking instead of prior passage reranking. Besides, the recently proposed pre-trained language model BERT is also applied to improve the performance of our model. Experiments with Chinese multi-passage dataset DuReader show that our model achieved competitive performance.

Acknowledgments

This work is supported by National Natural Science Foundation of China No.61751201, Research Foundation of Beijing Municipal Science and Technology Commission No. Z181100008918002. And we are greatful to Baidu Inc. and China Computer Federation for hosting competition and sharing data resources. We would also like to thank the anonymous reviewers for their insightful suggestions.

References

1. Yang, An, et al. "Adaptations of ROUGE and BLEU to Better Evaluate Machine Reading Comprehension Task." arXiv preprint arXiv:1806.03578 (2018).
2. Vaswani, Ashish, et al. "Attention is all you need." Advances in neural information processing systems. 2017.
3. Tan, Chuanqi, et al. "S-net: From answer extraction to answer generation for machine reading comprehension." arXiv preprint arXiv:1706.04815 (2017).
4. Clark, Christopher, and Matt Gardner. "Simple and effective multi-paragraph reading comprehension." arXiv preprint arXiv:1710.10723 (2017).
5. Hill, Felix, et al. "The goldilocks principle: Reading children's books with explicit memory representations." arXiv preprint arXiv:1511.02301 (2015).
6. Devlin, Jacob, et al. "Bert: Pre-training of deep bidirectional transformers for language understanding." arXiv preprint arXiv:1810.04805 (2018).
7. 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.
8. Hermann, Karl Moritz, et al. "Teaching machines to read and comprehend." Advances in neural information processing systems. 2015.

9. Nishida, Kyosuke, et al. "Multi-style Generative Reading Comprehension." arXiv preprint arXiv:1901.02262 (2019).
10. Nishida, Kyosuke, et al. "Retrieve-and-read: Multi-task learning of information retrieval and reading comprehension." Proceedings of the 27th ACM International Conference on Information and Knowledge Management. ACM, 2018.
11. Joshi, Mandar, et al. "Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension." arXiv preprint arXiv:1705.03551 (2017).
12. Dunn, Matthew, et al. "Searchqa: A new qa dataset augmented with context from a search engine." arXiv preprint arXiv:1704.05179 (2017).
13. Seo, Minjoon, et al. "Bidirectional attention flow for machine comprehension." arXiv preprint arXiv:1611.01603 (2016).
14. Yan, Ming, et al. "A Deep Cascade Model for Multi-Document Reading Comprehension." arXiv preprint arXiv:1811.11374 (2018).
15. Rajpurkar, Pranav, et al. "Squad: 100,000+ questions for machine comprehension of text." arXiv preprint arXiv:1606.05250 (2016).
16. Rajpurkar, Pranav, Robin Jia, and Percy Liang. "Know What You Don't Know: Unanswerable Questions for SQuAD." arXiv preprint arXiv:1806.03822 (2018).
17. Nguyen, Tri, et al. "MS MARCO: A human generated machine reading comprehension dataset." arXiv preprint arXiv:1611.09268 (2016).
18. He, Wei, et al. "DuReader: a Chinese Machine Reading Comprehension Dataset from Real-world Applications." Proceedings of the Workshop on Machine Reading for Question Answering. 2018.
19. Yang, Wei, et al. "End-to-End Open-Domain Question Answering with BERTserini." arXiv preprint arXiv:1902.01718 (2019).
20. Hu, Minghao, et al. "Retrieve, Read, Rerank: Towards End-to-End Multi-Document Reading Comprehension." arXiv preprint arXiv:1906.04618 (2019)
21. Wang, Shuohang, and Jing Jiang. "Machine comprehension using match-lstm and answer pointer." arXiv preprint arXiv:1608.07905 (2016).
22. Wang, Shuohang, et al. "R 3: Reinforced Ranker-Reader for Open-Domain Question Answering." Thirty-Second AAAI Conference on Artificial Intelligence. 2018.
23. Wang, Yizhong, et al. "Multi-passage machine reading comprehension with cross-passage answer verification." arXiv preprint arXiv:1805.02220 (2018).
24. Wang, Wenhui, et al. "Gated self-matching networks for reading comprehension and question answering." Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 2017.
25. Sun, Yu, et al. "ERNIE: Enhanced Representation through Knowledge Integration." arXiv preprint arXiv:1904.09223 (2019).