Scientific Reading Comprehension with Sentences Selection and Ranking

Jialei Chen[1], Weihua Wang[1,2,3(✉)], and Shuai Shao[1]

[1] College of Computer Science, Inner Mongolia University, Hohhot, China
32109166@mail.imu.edu.cn
[2] National and Local Joint Engineering Research Center of Intelligent Information Processing Technology for Mongolian, Hohhot, China
[3] Inner Mongolia Key Laboratory of Mongolian Information Processing Technology, Hohhot, China
wangwh@imu.edu.cn

Abstract. Scientific Machine Reading Comprehension (SMRC) aims to understand scientific long text by providing answers for the given questions. Most existing methods trend to answer the question using Transformer-based models. However, in the scientific domain, the original text is longer than the general domain. In this paper, we proposed a model that consists of a content retrieval module and a pre-trained model module. The content retrieval module finds the most semantically relevant sentences from the text and re-rank them. The seleted sentences and question will be input into the pre-trained model to get the answers. This model could overcome the length limitation of Transformer model length while achieving impressive results. Our model achieved 0.45 score of RougeL, resulting in the second place in the NLPCC2023 Shared Task2.

Keywords: Machine Reading Comprehension · Long Text Comprehension · Text retrieval · Pre-trained language model

1 Introduction

Machine Reading Comprehension (MRC) aims to enable machines to understand the content of texts, answer related questions, and provide accurate responses [1,2]. As a challenging research direction, MRC has gained significant interest from the research community due to its practical applications, such as question-answering dialogue systems [3,4]. The core challenge in MRC involves extracting pertinent information from a considerable amount of text, matching and reasoning with the information, and generating precise answers.

Scientific Machine Reading Comprehension (SMRC) is an extension of MRC specifically focused on comprehending and extracting text-specific information from scientific literature, academic papers, and other related texts to answer questions related to scientific knowledge [5]. NLPCC2023 Shared Task2 presents

F. Liu et al. (Eds.): NLPCC 2023, LNAI 14304, pp. 93–101, 2023.
https://doi.org/10.1007/978-3-031-44699-3_9

a multi-perspective scientific reading comprehension dataset that includes scientific papers and question-answering pairs from different perspectives. However, the substantial length of scientific papers presents a significant obstacle for reading comprehension.

To tackle the issue of lengthy texts in reading comprehension, we propose a simple retrieval and re-ranking method. Our approach is inspired by open-domain question-answering [6] and information retrieval techniques [7]. In the retrieval phase, our method treats sentences as fundamental text units and encodes them, along with the question, into two independent vector spaces using a bi-encoder. The encoded vector representations are then used to calculate cosine similarities, providing the K highest scoring sentences as candidate sets. In the subsequent re-ranking phase, a cross-encoder is employed to reorder combinations of the question and candidate sentences. The re-ordered sentences, along with the question, are input into a pre-trained model to generate the answer. This method effectively reduces text length, lowers computational costs, and captures relevant information scattered throughout long texts to answer questions.

Our retrieval and re-ranking pipeline significantly improves the baseline model by approximately 15 in terms of RougeL score on the NLPCC2023 Shared Task 2 dataset. The substantial enhancement in RougeL scores indicates that future research in scientific reading comprehension can benefit from the retrieval and re-ranking of text content that is more pertinent to the given questions.

2 Related Work

The QUALM system proposed by Lehnert [8] is an early MRC system, but due to its small scale, it has not been widely used. Hirschman [9] et al. proposed a bag-of-words technique that represents sentences with questions and context as sets of words and selects words that appear in both the question and the context as answers. Riloff [10] et al. designed a rule-based MRC system called Quarc, which contains different heuristic rules for different types of "wh" questions. Quarc also incorporates morphological analysis functions such as part-of-speech tagging, semantic categorization, and entity recognition. These works mainly rely on rule-based methods to solve MRC tasks. However, due to the lack of flexibility and the complexity of rule construction requiring expert knowledge, later works gradually shifted towards machine learning approaches. With the availability of large-scale benchmark datasets such as SQuAD [11], and Narrative QA [12], it became possible to tackle MRC tasks using deep neural architectures, providing a platform for evaluating the performance of MRC systems.

Subsequent work mainly focused on various attention-based interactions between passages and questions. Kadlec et al. [13] used Attention Sum, Dhingra et al. [14] used Gated attention, Wang et al. [15] used Self-matching, Cui et al. [16] used Attention over Attention, and Seo et al. [17] used Bi-attention. In our model, SentenceTransformer is used in the text retrieval part to calculate text similarity and to perform sentence ranking and retrieval based on the comparison of similarity between texts.

Recently, pre-trained language models (PrLMs) such as BERT [18], XLNet [19], and T5 [20] have achieved success in various natural language processing tasks. These powerful pre-trained encoders have demonstrated strong semantic encoding capabilities. Our model utilizes the T5 pre-trained model for encoding. These models typically consist of multiple layers of transformers, which encode sequences of limited length (e.g., 512). However, in some MRC tasks, input sequences may exceed the length limit. For example, documents in the TriviaQA dataset averagely contain 2,622 tokens. The SciMRC dataset used in this paper has an average of 6000 tokens per text. To handle documents with lengths far exceeding the input length of pre-trained models, it is common to split the documents into different segments and predict answers from each segment separately. The highest-scoring answer among these segments is selected as the final answer. Although this approach is simple, each part segmented from the document is treated independently, and the model fails to capture information that spans across text segments. Zhao et al. [21] predicted region answers from each block, compressed these answers into a new document, and then generated an answer. However, this two-step generation method inevitably introduces error propagation. Our model divides the text into individual sentences and selects sentences relevant to the question for answer generation. This approach only performs answer generation once, avoiding error propagation, and the sentence-level segmentation allows for finer extraction of textual information.

Regarding sentence retrieval, paragraph retrieval is an important step. Initially, TF/IDF-based sparse representation was used for retrieving supporting documents [22]. Lee et al. [23] introduced a supervised learning method that relies on BiLSTM to reorder paragraphs, while Wang et al. [24] trained a ranking system using reinforcement learning. The second approach to improve retrieval is by utilizing additional information such as Wikipedia or Wikidata graphs (Min et al. [25]; Asai et al. [26]). In our text retrieval, we use the Bi-encoder [27] approach for sentence selection, which is more efficient and performs better compared to previous methods.

3 Method

Our model draws inspiration from the human process of reading and comprehending lengthy texts. It focuses exclusively on the text content relevant to the given question. Our model comprises two components: the content retrieval model and the pre-trained model. The content retrieval model consists of two parts: retrieval and re-ranking, as depicted in Fig. 1. In the reading comprehension task, the scientific text is segmented into a collection of sentences. Both the sentence collection and the question collection are input into the content retrieval model. Specifically, for each question in the collection, a Bi-Encoder is employed to retrieve a candidate set from the sentence collection. Subsequently, re-ranking is performed to determine the sentence collection most semantically relevant to the given question. Finally, the question and the relevant sentence collection are fed into the pre-trained model for extracting the answer. This two-part model

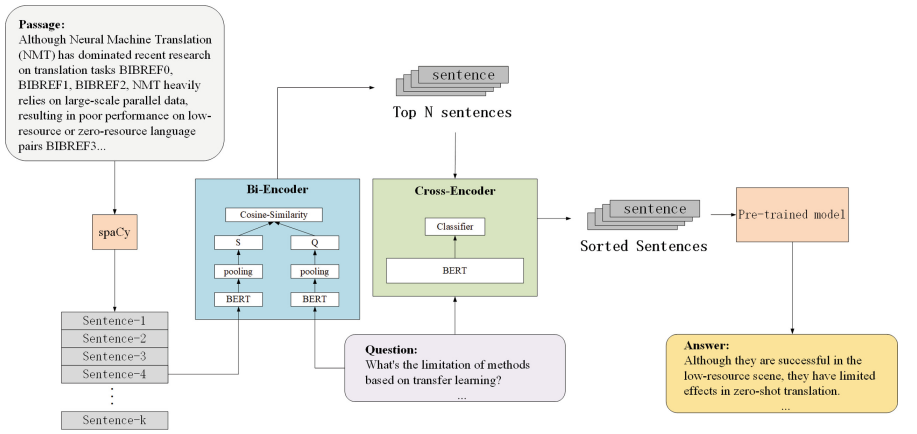

Fig. 1. The model consists of two parts: content retrieval and pre-trained model. The content retrieval part includes two steps: retrieval and re-ranking.

architecture efficiently retrieves and selects the most pertinent information for accurately answering the provided questions.

3.1 Retrieval Module

The retrieval component employs semantic search, which differs from traditional search methods reliant on lexical matching. Semantic search matches items based on their semantic information. The approach involves embedding all entries in the corpus, be it sentences, paragraphs, or documents, into a vector space. During the search process, the query is also embedded into the same vector space, and the closest embedding is identified from the corpus. These selected entries are expected to exhibit a high semantic overlap with the query. In our implementation, the sentence collection obtained by segmenting the articles in the dataset serves as the corpus, while the questions serve as queries for the semantic search process.

The bi-encoder is a commonly used model architecture in natural language processing tasks such as semantic search, sentence similarity calculation, and question-answering systems. It is designed to generate semantic vectors for text pairs. The bi-encoder architecture involves independently encoding the query text and candidate text (typically from a document collection or question-answer database) into two separate semantic vectors, without sharing parameters between them. The generation of semantic vectors follows an encoder-decoder structure. The encoder transforms an input text into a fixed-dimensional semantic vector that captures its semantic information. Similarity metrics such as cosine similarity or Euclidean distance can then be used to compute the similarity between two vectors. By utilizing the semantic vectors from the bi-encoder, it is possible to calculate similarity scores between queries and candidate texts

in semantic search and find the most relevant texts to a query. The bi-encoder architecture is also applicable to tasks like sentence similarity calculation, where the semantic vectors of two sentences are compared to assess their similarity. The advantages of the bi-encoder architecture include high computational efficiency, as the encoder only needs to compute the semantic vectors once for each query, enabling efficient search and similarity calculations on large-scale text collections.

To input a collection of questions $Q = \{q_1, ...q_n\}$ and a collection of sentences $S = \{s_1, ...s_m\}$ into the bi-encoder, we obtain representations of the question and the sentence in the same vector space. Then, we calculate the semantic similarity between the question and the sentence using cosine similarity and select the top k highest cosine similarity scores as the retrieve candidates, where k is a hyperparameter.

3.2 Re-Ranking Module

Re-Rank is the process of rearranging the retrieve candidate sentences using a cross-encoder. The bi-directional encoder in the retrieval part captures some semantic information, but it mainly reduces the computational cost of traditional encoders in large-scale corpus matching.

The cross-encoder is a model architecture used to generate semantic vectors, which is employed in various natural language processing tasks such as text classification, sentence similarity calculation, and question-answering systems. Unlike the bi-encoder, the cross-encoder compares the combination of query and candidate texts when generating semantic vectors. It takes the query and candidate texts as input, processes them in the encoder, and generates an overall semantic vector representation. The design principle of the cross-encoder is to better capture the semantic relationship between queries and candidate texts. This can be achieved by using attention mechanisms, transformers, or other neural network structures. Compared to the bi-encoder, the cross-encoder model has higher computational costs because it requires encoding each query-candidate text pair. The semantic vectors generated by the cross-encoder can be used in text classification tasks to determine the degree of matching between queries and candidate texts or in sentence similarity calculation to assess the similarity between two sentences. In question-answering systems, the cross-encoder can compare the question with possible answer options to find the most relevant answer. The advantage of the cross-encoder lies in its ability to comprehensively consider the semantic relationship between queries and candidate texts, but it incurs higher computational costs due to the need for comparing every text pair. Therefore, in tasks requiring deeper semantic understanding, the cross-encoder model can provide more accurate matching and similarity calculation results.

4 Experiments

In this section, we introduce the dataset, baseline model, implementation details, evaluation metrics, and key results of this task.

4.1 Setuping

We implement sentence tokenization using Spacy, an open-source natural language processing (NLP) library designed for efficient text processing. Built on Python, Spacy provides a range of tools and functions for text processing.

The neural network is implemented using PyTorch, SentenceTransformer, and the HuggingFace Transformers library. For the content retrieval part, we employ the all-MiniLM-L6-v2 model from SentenceTransformer. This model is obtained by compressing a large pre-trained Transformer-based language model using knowledge distillation techniques, and it follows a bi-encoder architecture.

For the re-ranking part, we utilize the ms-marco-TinyBERT-L-2 model. This model is obtained by compressing a large pre-trained BERT-based language model using knowledge distillation techniques, and it follows a cross-encoder architecture.

For answer generation, we employ the T5-base model. During the training of the model, we set the learning rate to 1e–4, the batch size to 16, and train for 10 epochs. The maximum input length is set to 512, and the maximum output length is set to 128.

We implemented the model in PyTorch and trained it on a single NVIDIA A40 GPU with 48 GB VRAM. In our implementation, we set the hyperparameter k to 10, which means that we selected the top 10 most similar sentences from the article based on the bi-encoder and cosine similarity calculation.

4.2 Tasks and Datasets

We evaluate our model on the dataset provided by the NLPCC2023 ShareTask2. The training and validation sets consist of 592 data samples, where each sample contains five attributes: title, abstract, fulltext, figures, tables, and qas. These sets include a total of 4,873 question-answer pairs. The test set consists of 1,169 question-answer pairs.

4.3 Evaluation Metrics

We evaluated our model on the dataset provided by the NLPCC2023 ShareTask2. The training and validation sets consisted of 592 data samples, each containing five attributes: title, abstract, fulltext, figures, tables, and qas. These sets included a total of 4,873 question-answer pairs. The test set consisted of 1,169 question-answer pairs.

$$\begin{aligned} R_{lcs} &= \frac{LCS(X,Y)}{m} \\ p_{lcs} &= \frac{LCS(X,Y)}{n} \\ F_{lcs} &= \frac{(1+\beta^2)R_{lcs}P_{lcs}}{R_{lcs}+\beta^2 P_{lcs}} \end{aligned} \tag{1}$$

4.4 Main Results

The experimental results of our proposed model and the baseline model are shown in the table1. From the results, we can observe that our proposed model outperforms the baseline model significantly in various aspects, including Rouge and Bleu scores. Using RougeL as the evaluation metric, our model achieved a score of 45.

In order to further analyze the effectiveness of our model, we conducted ablation experiments to demonstrate the effectiveness of the content retrieval module. These experiments were performed on the train & validation dataset, which was divided into training, validation, and testing sets in an 8:1:1 ratio. The experimental parameters were kept consistent throughout, and we conducted experiments on both the t5-base and t5-small models.

As shown in the Table 2, when the content retrieval module was removed, the rougeL scores decreased by approximately 11.5 across different sizes of pre-training models. This indicates that by providing text content that is semantically relevant to the question, we are able to generate decent answers. The use of the entire text as input may prevent the model from focusing on the main content, resulting in noise instead.

Table 1. Experiment results on NLPCC2023 SharedTask2 dataset.

model	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-Lsum
Baseline	32.82	19.02	**30.58**	30.55
Ours	47.14	35.30	**45.18**	45.32

Table 2. Rouge&Bleu-score of ablation experiments for pre-trained models.

model	ROUGE-1	ROUGE-2	ROUGE-L	BLEU1	BLEU4
Baseline(T5-small)	28.64	15.31	**25.94**	17.24	6.18
Ours(T5-small)	39.49	27.37	**37.44**	12.22	6.01
Baseline(T5-base)	32.82	19.02	**30.58**	16.62	6.76
Ours(T5-base)	47.14	35.30	**45.18**	14.35	7.24

5 Conclusion

This paper introduces our model, which is capable of handling long-text scientific question-answering tasks. Our model can extract the sentences relevant to the question from long texts and effectively answer different questions based on these relevant sentences. Experimental results show that the model is able to identify and infer correct answers in long texts when facing different perspectives from beginners, students, and experts. The final model achieved the second-best

results on the SciMRC dataset. It is important to note that although the SciMRC dataset is related to the field of science, our model is also suitable for other domains. For future work, we hope to improve the model's ability to compute the similarity between questions and text and enhance its ability to extract useful information from the text. We acknowledge that the selection of sentences by the model is crucial, as discarding sentences containing key information can significantly impact the model's results. Therefore, sentence selection becomes particularly critical for the model.

Acknowledgment. This work is supported by National Natural Science Foundation of China (Nos. 62066033, 61966025); Inner Mongolia Applied Technology Research and Development Fund Project (Nos. 2019GG372, 2020PT0002, 2022YFDZ0059); Inner Mongolia Natural Science Foundation (2020BS06001). We are grateful for the useful suggestions from the anonymous reviewers.

References

1. Chen, D.: Neural Reading Comprehension and Beyond. Stanford University, Stanford (2018)
2. Hermann, K.M., et al.: Teaching machines to read and comprehend. In: Advances in Neural Information Processing Systems, vol. 28 (2015)
3. Wen, T.H., et al.: A network-based end-to-end trainable task-oriented dialogue system. arXiv preprint arXiv:1604.04562 (2016)
4. Chen, H., Liu, X., Yin, D., Tang, J.: A survey on dialogue systems: Recent advances and new frontiers. ACM SIGKDD Explor. Newsl. **19**(2), 25–35 (2017)
5. Zhang, X., Zheng, H., Nie, Y., Huang, H., Mao, X.L.: SCIMRC: multi-perspective scientific machine reading comprehension. arXiv preprint arXiv:2306.14149 (2023)
6. Chen, D., Yih, W.T.: Open-domain question answering. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts, pp. 34–37 (2020)
7. Manning, C.D.: An Introduction to Information Retrieval. Cambridge University Press, Cambridge (2009)
8. Lehnert, W.G.: The Process of Question Answering. Yale University, New Haven (1977)
9. Hirschman, L., Light, M., Breck, E., Burger, J.D.: Deep read: a reading comprehension system. In: Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics, pp. 325–332 (1999)
10. Riloff, E., Thelen, M.: A rule-based question answering system for reading comprehension tests. In: ANLP-NAACL 2000 Workshop: Reading Comprehension Tests as Evaluation for Computer-Based Language Understanding Systems (2000)
11. Rajpurkar, P., Zhang, J., Lopyrev, K., Liang, P.: Squad: 100,000+ questions for machine comprehension of text. arXiv preprint arXiv:1606.05250 (2016)
12. Joshi, M., Choi, E., Weld, D.S., Zettlemoyer, L.: TriviaQA: a large scale distantly supervised challenge dataset for reading comprehension. arXiv preprint arXiv:1705.03551 (2017)
13. Kadlec, R., Schmid, M., Bajgar, O., Kleindienst, J.: Text understanding with the attention sum reader network. arXiv preprint arXiv:1603.01547 (2016)
14. Dhingra, B., Liu, H., Yang, Z., Cohen, W.W., Salakhutdinov, R.: Gated-attention readers for text comprehension. arXiv preprint arXiv:1606.01549 (2016)

15. Wang, W., Yang, N., Wei, F., Chang, B., Zhou, M.: Gated self-matching networks for reading comprehension and question answering. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 189–198 (2017)
16. Cui, Y., Chen, Z., Wei, S., Wang, S., Liu, T., Hu, G.: Attention-over-attention neural networks for reading comprehension. arXiv preprint arXiv:1607.04423 (2016)
17. Seo, M., Kembhavi, A., Farhadi, A., Hajishirzi, H.: Bidirectional attention flow for machine comprehension. arXiv preprint arXiv:1611.01603 (2016)
18. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
19. Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R.R., Le, Q.V.: XLNet: generalized autoregressive pretraining for language understanding. In: Advances in Neural Information Processing Systems, vol. 32 (2019)
20. Raffel, C., et al.: Exploring the limits of transfer learning with a unified text-to-text transformer. J. Mach. Learn. Res. **21**(1), 5485–5551 (2020)
21. Zhao, J., et al.: ROR: read-over-read for long document machine reading comprehension. arXiv preprint arXiv:2109.04780 (2021)
22. Chen, D., Fisch, A., Weston, J., Bordes, A.: Reading wikipedia to answer open-domain questions. arXiv preprint arXiv:1704.00051 (2017)
23. Lee, J., Yun, S., Kim, H., Ko, M., Kang, J.: Ranking paragraphs for improving answer recall in open-domain question answering. arXiv preprint arXiv:1810.00494 (2018)
24. Wang, S., et al.: R 3: reinforced ranker-reader for open-domain question answering. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 32 (2018)
25. Min, S., Chen, D., Zettlemoyer, L., Hajishirzi, H.: Knowledge guided text retrieval and reading for open domain question answering. arXiv preprint arXiv:1911.03868 (2019)
26. Asai, A., Hashimoto, K., Hajishirzi, H., Socher, R., Xiong, C.: Learning to retrieve reasoning paths over wikipedia graph for question answering. arXiv preprint arXiv:1911.10470 (2019)
27. Yi, X., et al.: Sampling-bias-corrected neural modeling for large corpus item recommendations. In: Proceedings of the 13th ACM Conference on Recommender Systems, pp. 269–277 (2019)