Multi-Granularity Matching Network for Multi-Paragraph Machine Reading Comprehension

Xu Chen
University of Chinese Academy of Sciences, Beijing, 100000, China
Corresponding author’s e-mail: chenxu18@mails.ucas.ac.cn

Abstract. Multi-paragraph aims to allow machines to read multiple paragraphs and infer the answer to the given question. Usually, the model needs to use the selector to narrow the range of candidate paragraphs, and then use the reader to find the answer. For the selector, the previous work generally matches the question and the paragraph on the word level and the paragraph level. In this paper, we study the encoding and matching algorithm at the chunk level, which is supposed to make our model more accurate. Combining it with gated self matching mechanism, we design a multi-granularity matching network as the selector. The experimental results show that our model achieves competitive performance on Quasar-T.

1. Introduction
Machine reading comprehension (MRC) aims to read paragraphs by machines and give appropriate answers to questions designed by humans. According to the different answer forms, it can be divided into multiple-choice, cloze-style, span extraction, and free form. In recent years, multi-paragraph MRC has received a lot of attention, which is an extension of MRC, but at this time the machine needs to read multiple paragraphs. Compared with single-paragraph MRC, multi-paragraph MRC is more practical. Since it usually follows the Retrieve-then-Read pattern, it is regarded as a combination of information retrieval technology and MRC technology. This paper focuses on multi-paragraph span extraction MRC.

Generally, the multi-paragraph MRC model consists of two steps: firstly, the selector is used to filter out the paragraphs most likely to contain the answer in the candidate paragraphs, and then the reader uses the MRC model to find the most suitable span to answer the question [1]. Many models have followed this structure, and use deep neural networks to achieve excellent results on this task.

For the retriever, previous work shows that it is effective to match the question and paragraph at different granularity, such as character level, word level, and paragraph level [2][3]. However, they ignore the granularity of combining adjacent words. For example, by treating every three adjacent words as a whole to match the question and the paragraph, we can select paragraphs that are similar to the question sentence in triple granularity. The local words encoding process can be implemented by one-dimensional CNN, and we use chunk to represent each local representation after encoding. We developed a new selector based on this idea.

Our contributions are as follows:
- We propose to match the question and paragraph at the chunk level formed by local connection, which enables the model to capture their common features under different word combinations.
We propose to match the question and paragraph at the chunk level formed by local connection, which enables the model to capture their common features under different word combinations.

Experiments prove that our model is competitive.

2. Related Work
Multi-paragraph MRC model usually contains two modules, selector and reader. The former is responsible for locating paragraphs that may contain answers in multiple documents, and the latter applies the MRC model to find answers. For MRC, reference [4] first proposed a deep neural network based on attention to solve this problem. Reference [5] combined match-LSTM and pointer net, and [6] used bidirectional attention to obtain a question-aware context representation. Reference [7] proposed reader using a shared-norm objective function for multiparagraph MRC.

For selector, reference [1] applied the IR system to select relevant paragraphs. Reference [8] used a deep cascade model to keep the balance between effectiveness and efficiency by reading the candidate documents in a coarse-to-fine manner. Reference [9] developed a gated token selection mechanism to select important tokens, reference [2] proposed an improved attention-based model which makes full use of the question-paragraph and the paragraph-paragraph relevance.

Some models add a third module called re-ranker or verifier, which is used to re-rank or verify candidate answers. Reference [10] proposed strength-based and coverage-based re-ranking strategy to utilize the aggregated evidence from different paragraphs. Reference [11] developed a cross-passage answer verification module to make the candidate answer exchange information and verify each other.

Recently, transformer-based pre-training models such as BERT [12], ALBERT [13], etc. have shown dominance on multiple NLP tasks. They usually perform pre-training on massive text data and then fine-tune them on downstream tasks. Because our experiment did not use additional datasets, for fairness, we do not compare with models that use pre-trained models.

3. Model
3.1. Task Definition
For the artificially designed question \( Q \), given a set of candidate paragraphs \( P = \{P_i, i = 1, 2, 3, ... L\} \), where \( L \) is the number of paragraphs, the model needs to score the relevance of candidate paragraphs to the question, and then input the high-scoring paragraphs into the MRC module to find the answer \( A \).

3.2. Selector
Our work mainly focuses on selector. As shown in Figure 1, our selectors match paragraphs and questions from word level, chunk level and paragraph level.

3.2.1. Embedding Layer. As shown in the figure, in the embedding layer, both \( P \) and \( Q \) need to go through character embedding and word embedding, and then concatenate the output to get \( c \) and \( q \) respectively. Character embedding consists of a one-dimensional convolutional layer and a max-pooling layer, which are learnable, while the word embedding layer is fixed and initialized with pre-trained word embedding.

3.2.2. Pre-encoder Layer. We use BiLSTM to perform context encoding on paragraph and question to get \( H_c \in \mathbb{R}^{d \times |c|} \) and \( H_q \in \mathbb{R}^{d \times |q|} \), where \( d \) represents the hidden size, \( |c| \) and \( |q| \) represent the length of \( c \) and \( q \) respectively:

\[
H_c = \text{BiLSTM}(c), \quad H_q = \text{BiLSTM}(q)
\]
Figure 1. The architecture of our selector.

Then we use the same bidirectional attention flow layer as in [6], [7] to get the question-aware contextual representation $S \in \mathbb{R}^{d \times |S|}$. Due to space limitations, we will not describe it in detail.

3.2.3. Gated self-matching Layer. To obtain the interaction relationship between the various words of the entire paragraph, and to strengthen the representation of the word related to the question, we adopt the gated self-matching mechanism [14]. First, we calculate the self-matching representation $u_i$:

$$a_i = \text{softmax}(S^T W^S S_i), \quad u_i = S \cdot a_i$$ (2)

where $a_i$ is the self-attention weight. Then we calculate the updated representation $m_i$ and the gating signal $g_i$ to get $O_i$:

$$m_i = \tanh(W^m [S_i; u_i]), \quad g_i = \text{sigmoid}(W^g [S_i; u_i])$$

$$O_i = g_i \odot m_i + (1 - g_i) \odot S$$ (4)

where $W^S$, $W^m$, $W^g$ are all learnable matrices, $\odot$ is the element-wise multiplication.

3.2.4. Chunk level matching layer. To obtain the representation of the local word block, that is, the chunk level representation, we use a two-layer one-dimensional convolutional neural network following the tanh activation function:

$$x^0 = \tanh(\text{Conv}_0(O)), \quad y^0 = \tanh(\text{Conv}_0(H_y))$$

$$x^1 = \tanh(\text{Conv}_1(x^0)), \quad y^1 = \tanh(\text{Conv}_1(y^0))$$ (6)

where $x$ and $y$ are representations of paragraph and question, respectively. Then we match the paragraph with the question again to obtain the vector representation of the paragraph $V_x$:

$$E = x \cdot W^E \cdot y, \quad \beta_i = \text{softmax} \left( \sum_j E_{ij} \right), \quad V_x = \sum_i \beta_i \cdot x$$ (7)
where $E$ is the similarity matrix, and $\beta_i$ represents the weight of the i-th chunk. We get the vector representation of the question in a similar way, but this time the question matches itself:

$$ R = y \cdot W^\beta \cdot y, \quad \alpha_i = \text{softmax} \left( \sum_j R_{yj} \right), \quad V_y = \sum_i \alpha_i \cdot y, $$(8)

### 3.2.5. Paragraph level matching layer.
We use $Z \in \mathbb{R}^{d \cdot L}$ to represent all candidate paragraphs’ vector representation, and then calculate the score for each paragraph:

$$ Score = \text{softmax} \left( Z \cdot V_y \right) $$ (9)

### 3.3. Reader
In our model, selector and reader are two independent modules, so the reader can be replaced with any MRC model. We use the same reader as [7], and use a sampling strategy similar to [2], that is, weighted sampling of candidate paragraphs according to the output of the selector, which can make the reader less sensitive to the results of the selector and make the model more robust.

### 4. Experiment

#### 4.1. Dataset
We conduct experiments on Quasar-T [15] to verify the performance of the model. Quasar-T contains 43,000 open-domain question-answer pairs, and each question has 100 processed passes retrieved from the Internet. Candidate paragraphs are not necessarily related to the question or contain answer span. It is worth noting that the answers to some questions do not exist in any candidate passage.

#### 4.2. Experimental Setting
In our model, the hidden size is set to 150. The kernel sizes of the two layers of 1-d CNN in the Section 3.2.4 are 1 and 3 respectively. In the embedding layer, we use pre-trained 300-d GloVe word vectors. The batch size is set to 32, the Adam algorithm is used for optimization, and the learning rate is set to 5e-4.

| Model          | EM  | F1  | h@1 | h@3 | h@5 |
|----------------|-----|-----|-----|-----|-----|
| BiDAF [6]      | 25.9| 28.5| -   | -   | -   |
| R3 [16]        | 35.3| 41.7| 40.3| 51.3| 54.5|
| DS-QA [17]     | 37.3| 43.7| 27.7| 36.8| 42.6|
| Shared-Norm [7]| 38.6| 45.4| -   | -   | -   |
| DynSAN [9]     | 48.0| 54.8| -   | -   | -   |
| RASA [2]       | 48.6| 57.8| 49.9| 58.8| 62.6|
| Our Model      | 48.1| 56.2| 49.3| 58.5| 62.5|

| Model                  | h@1 | h@3 | h@5 |
|------------------------|-----|-----|-----|
| Full Model              | 49.3| 58.5| 62.5|
| -BiDAF                  | 45.9| 52.4| 57.2|
| -Gated self-matching    | 47.6| 54.8| 59.7|
| -chunk level encoding   | 48.5| 57.0| 61.6|
4.3. Results
We compare our model with several representative models, and the experimental results are shown in the Table 1. Among them, h@1, h@3, and h@5 are used to evaluate the performance of the selector, which respectively represent the proportion of questions that contain answers in the top 1, 3, and 5 documents. EM and F1 use the answers output by the reader to evaluate the performance of the entire model, including selector and reader. It can be seen that the performance of our model exceeds most of the great models, and it is also competitive compared with the SOTA model.

4.4. Ablation Study
We conducted ablation experiments on Quasar-T and the results are shown in the Table 2. The BiDAF module has a great influence on the model, which means that matching paragraphs with question at word level is very important. The gated self-matching module also played a positive role in the model. The performance of the model has a significant decline while we remove the two-layer one-dimensional CNN in the chunk level matching module, which shows that one-dimensional CNN is effective for extracting local relations.

5. Conclusion
In this paper, we propose a novel selector for multi-paragraph MRC, which matches candidate paragraphs with the question from multiple granularities. Compared with previous work, we study the similarity between paragraphs and question at the chunk level and use one-dimensional CNN to model the local relationship of the text. Based on the chunk level representation, we design a matching network. Experimental results prove that our improvement is effective, and our model demonstrates a competitive performance on Quasar-T.

References
[1] D. Chen, A. Fisch, J. Weston, and A. Bordes. (2017) Reading wikipedia to answer open-domain questions. http://arxiv.org/abs/1704.00051
[2] Y. Xu, Z. Lin, Y. Liu, R. Liu, W. Wang, and D. Meng. (2019) Ranking and sampling in open-domain question answering. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). Hong Kong. pp. 2412-2421.
[3] M. Hu, Y. Peng, Z. Huang, and D. Li. (2019) Retrieve, read, rerank: Towards end-to-end multi-document reading comprehension. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. Florence. pp. 2285-2295.
[4] K. M. Hermann, T. Kocisky, E. Grefenstette, L. Espeholt, W. Kay, M. Suleyman, and P. Blunsom. (2015) Teaching machines to read and comprehend. In: Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems. Montreal, Quebec, Canada, pp. 1693-1701.
[5] S. Wang and J. Jiang. (2016) Machine comprehension using match-lstm and answer pointer. http://arxiv.org/abs/1608.07905
[6] M. J. Seo, A. Kembhavi, A. Farhadi, and H. Hajishirzi. (2017) Bidirectional attention flow for machine comprehension. In: 5th International Conference on Learning Representations. Toulon, France.
[7] C. Clark and M. Gardner. (2018) Simple and effective multi-paragraph reading comprehension. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics. Melbourne. pp. 845-855.
[8] M. Yan, J. Xia, C. Wu, B. Bi, Z. Zhao, J. Zhang, L. Si, R. Wang, W. Wang, and H. Chen. (2019) A deep cascade model for multi-document reading comprehension. In: Proceedings of the AAAI Conference on Artificial Intelligence. Honolulu. pp. 7354-7361.
[9] Y. Zhuang and H. Wang. (2019) Token-level dynamic self-attention network for multi-passage reading comprehension. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. Florence. pp. 22522262.

[10] S. Wang, M. Yu, J. Jiang, W. Zhang, X. Guo, S. Chang, Z. Wang, T. Klinger, G. Tesauro, and M. Campbell. (2018) Evidence aggregation for answer re-ranking in open-domain question answering. In: 6th International Conference on Learning Representations. Vancouver.

[11] Y. Wang, K. Liu, J. Liu, W. He, Y. Lyu, H. Wu, S. Li, and H. Wang. (2018) Multi-passage machine reading comprehension with cross-passage answer verification. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics. Melbourne. pp.19181927.

[12] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. (2019) Bert: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics. Minneapolis. pp. 41744186.

[13] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut. (2020) ALBERT: A lite BERT for self-supervised learning of language representations. In 8th International Conference on Learning Representations. Addis Ababa.

[14] W. Wang, N. Yang, F. Wei, B. Chang, and M. Zhou. (2017) Gated selfmatching networks for reading comprehension and question answering. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics. Vancouver. pp. 189198

[15] B. Dhingra, K. Mazaitis, and W. W. Cohen. (2017) Quasar: Datasets for question answering by search and reading. http://arxiv.org/abs/1707.03904

[16] S. Wang, M. Yu, X. Guo, Z. Wang, T. Klinger, W. Zhang, S. Chang, G. Tesauro, B. Zhou, and J. Jiang. (2018) R 3: Reinforced ranker-reader for open-domain question answering. In: Thirty-Second AAAI Conference on Artificial Intelligence. New Orleans. pp. 5981-5988.

[17] Y. Lin, H. Ji, Z. Liu, and M. Sun. (2018) Denoising distantly supervised open-domain question answering. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics. Melbourne. pp. 17361745.