

Deep Understanding based Multi-Document Machine Reading Comprehension

FEILIANG REN, Northeastern University, China
 YONGKANG LIU, Northeastern University, China
 BOCHAO LI, Northeastern University, China
 ZHIBO WANG*, Northeastern University, China
 YU GUO*, Northeastern University, China
 SHILEI LIU*, Northeastern University, China
 HUIMIN WU*, Northeastern University, China
 JIAQI WANG*, Northeastern University, China
 CHUNCHAO LIU*, Northeastern University, China
 BINGCHAO WANG*, Northeastern University, China

Most existing multi-document machine reading comprehension models mainly focus on understanding the interactions between the input question and documents, but ignore following two kinds of understandings. First, to understand the semantic meaning of words in the input question and documents from the perspective of each other. Second, to understand the supporting cues for a correct answer from the perspective of intra-document and inter-documents. Ignoring these two kinds of important understandings would make the models oversee some important information that may be helpful for finding correct answers. To overcome this deficiency, we propose a deep understanding based model for multi-document machine reading comprehension. It has three cascaded deep understanding modules which are designed to understand the accurate semantic meaning of words, the interactions between the input question and documents, and the supporting cues for the correct answer. We evaluate our model on two large scale benchmark datasets, namely TriviaQA Web and DuReader. Extensive experiments show that our model achieves state-of-the-art results on both datasets.

CCS Concepts: • **Information systems** → **Question answering**.

Additional Key Words and Phrases: question and answering, multi-document machine reading comprehension, accurate word semantic meaning understanding, interaction understanding, answer supporting cue understanding, DuReader, TriviaQA Web

ACM Reference Format:

Feiliang Ren, Yongkang Liu, Bochao Li, Zhibo Wang, Yu Guo, Shilei Liu, Huimin Wu, Jiaqi Wang, Chunchao Liu, and Bingchao Wang. 2022. Deep Understanding based Multi-Document Machine Reading Comprehension. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.* 1, 1, Article 1 (January 2022), 21 pages. <https://doi.org/10.1145/3519296>

*These authors contribute equally to this research and are listed randomly.

Authors' addresses: Feiliang Ren, renfeiliang@cse.neu.edu.cn, Northeastern University, 11 Wenhua Rd, Heping Qu, Shenyang City, China; Yongkang Liu, Northeastern University, China; Bochao Li, Northeastern University, China; Zhibo Wang, Northeastern University, China; Yu Guo, Northeastern University, China; Shilei Liu, Northeastern University, China; Huimin Wu, Northeastern University, China; Jiaqi Wang, Northeastern University, China; Chunchao Liu, Northeastern University, China; Bingchao Wang, Northeastern University, China.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2022 Copyright held by the owner/author(s). Publication rights licensed to ACM.

2375-4699/2022/1-ART1 \$15.00
<https://doi.org/10.1145/3519296>

1 INTRODUCTION

Machine reading comprehension (MRC) aims to answer questions by reading given documents. It is considered one of the core abilities of artificial intelligence (AI) and the foundation of many AI-related applications like next-generation search engines and conversational agents. In real-world scenarios, MRC is often required to answer questions based on multiple documents. So multi-document MRC is receiving growing research interests [Clark and Gardner 2018; Hu et al. 2019; Joshi et al. 2017; Peng et al. 2020; Yan et al. 2019; Zemlyanskiy et al. 2021].

Generally, there are following three main challenges in the multi-document MRC. (i) It requires a model have the ability of processing very long text. For example, in TriviaQA Web [Joshi et al. 2017], a benchmark multi-document MRC dataset, there are averagely about 7 documents for each question in its training set, and each document contains averagely about 2,895 words. In DuReader [He et al. 2018], another benchmark multi-document MRC dataset, there are about 5 documents for each question, and each document contains averagely about 1,793 Chinese characters. In contrast, in SQuAD [Rajpurkar et al. 2018], a benchmark single-document MRC dataset, there is only one document for each question, and each document contains averagely about 735 words. (ii) In the multi-document MRC, there are many distractors of an answer: some spans have very high lexical matching results with the answer but completely different semantic meaning with the answer. Thus it requires a model have the ability of accurately understanding the semantic meaning of words in a document and its corresponding question. (iii) The location of an answer is very flexible in the multi-document MRC: it may appear once or multiple times in only a document, and it may also appear multiple times in multiple documents. Obviously, this kind of information is useful for finding correct answers by mutual authentication from following two aspects. (i) If a text span (not some meaningless function words) appears repeatedly in the input documents, it would be highly possible to be related to the correct answers; (ii) If a text span only appears once in only a document, it would be less possible to be related to the correct answer. Thus it requires a model have the ability of mining such kind of information accurately.

Although these challenges are difficult to handle, researchers notice that human readers can well overcome them by using some reading patterns like the patterns of “*read + verify*” or *multi-step reasoning*. Inspired by this, researchers begin to imitate human’s reading patterns when they design MRC models and lots of novel multi-documents MRC models are proposed [Chen et al. 2020; Clark and Gardner 2018; Hu et al. 2019; Malmaud et al. 2020; Peng et al. 2020; Tian et al. 2020; Wang et al. 2018c; Yan et al. 2019; Zhang et al. 2021]. Experiments show that these imitations are very effective and the corresponding models achieve state-of-the-art results on many benchmark datasets.

However, most of these existing methods pay more attention to a reading pattern’s *superficial frameworks*, which means they are prone to design a model that have the same or similar processing steps as a human’s reading pattern. For example, if they imitate human’s “*read + verify*” reading pattern, then they are prone to design a *read* module and a *verify* module in their MRC model. Similarly, if they simulate human’s multi-step reasoning pattern, then they are prone to design an iterative-style MRC model. The main deficiency of these existing models is that they ignore the underlying motivations of human readers using diverse reading patterns are to comprehensively *understand* the semantic meaning of the given documents and questions. Some researchers [Gong et al. 2020a; Guo et al. 2020b; Mihaylov and Frank 2019; Zhang et al. 2020a] explore the semantic information understanding issue, but their methods either require some prerequisite resources like an extra knowledge base [Guo et al. 2020b] or the linguistic annotations [Mihaylov and Frank 2019], or depend on some large scale pretrained language models [Zhang et al. 2020a].

We further notice that there are usually three kinds of *hierarchical understandings* when human readers conduct a reading comprehension task, including the *semantic meaning understanding of*

Table 1. An example extracted from TriviaQA Web. The answer is in **bold** and the key information is in *italic*.

Q	Which volcano in Tanzania s the highest mountain in Africa?
A	Mount Kilimanjaro
P1	Mount Kilimanjaro , <i>the Highest Volcano in Tanzania , Africa</i> World Tourism Place \n Stunning Views \n Mount Kilimanjaro , <i>the Highest Volcano in Tanzania , Africa...</i> is one of <i>the highest volcanoes in the world and is the highest mountain in Africa...</i>
P2	Mount Kilimanjaro - <i>Tanzania Africa</i> - YouTube \n ... Welcome to Mount Kiliman- jaro a dormant volcano which is <i>the highest mountain in Africa... in Tanzania , Africa...</i>
P3	... Sunrise on Mount Kilimanjaro . \n © Anna Omelchenko/Fotolia \n A caldera on Kibo , Mount Kilimanjaro...
P4	... Where is Mount Kilimanjaro \n The cloud-swathed peaks of <i>Africa ' s highest mountain...</i>

words, the *interaction understanding* between the input question and documents, and the *answer supporting cue understanding* among different documents. Most existing models focus on designing attention based methods for the *interaction understanding* and designing a simple embedding layer for the *semantic meaning understanding of words*, but paying less attention to the *answer supporting cue understanding*. We call these existing methods as shallow *understanding* based models, and they usually suffer from following two deficiencies. First, these models could not accurately *understand* the semantic meaning of words. In the MRC task, the input question and documents are deeply correlated. Thus their words' semantic meaning should not be understood in isolation. Especially when the input question and documents contain out-of-vocabulary (OOV) words, polysemy phenomenon, and synonymy phenomenon. Second, these models do not make full use of the information provided by documents. Usually, a question's given documents have similar semantic meaning, and the answer may occur in some of them or appear many times in one of them. All such information is helpful for finding the answer and should be fully used.

To address these two deficiencies, we propose a *deep understanding* based multi-document MRC model. The core idea of our method can be briefly illustrated by the example demonstrated in Table 1. In this example, even if "*Tanzania*" in the question is an OOV word, its semantic meaning can still be well understood when using the given documents as context since there is much key information available for understanding its accurate semantic meaning. For example, the context "...*the Highest Volcano in ...*" and "...*the highest mountain in ...*" occur many times around "*Tanzania*", which indicates that "*Tanzania*" is highly possible to be a location. Besides, "*Mount Kilimanjaro*" occurs many times in a document and many documents talk about it, both increase the probability of it being the answer.

Specifically, the proposed model contains three cascaded deep understanding modules to imitate human's three kinds of *understandings*. Besides the widely discussed *interaction understanding*, our model can also understand: (i) the semantic meaning of words by placing them into some specific contexts: taking documents as context when *understanding* the semantic meaning of a word in the question, and taking the question as context when *understanding* the semantic meaning of a word in documents, and (ii) the answer supporting cues by mining features from the aspects of intra-document and inter-document.

We evaluate our model on two large-scale multi-document MRC benchmark datasets, TriviaQA Web [Joshi et al. 2017] and DuReader [He et al. 2018]. Extensive experiments show that the proposed model is very effective and it achieves competitive results on both of them.

2 RELATED WORK

According to the number of documents given for a question, we categorize the MRC task into single-document MRC and multi-document MRC.

2.1 Single-document MRC

Based on the work of [Nishida et al. 2019; Seo et al. 2016; Yu et al. 2018], etc, we classify the main modules in the models of this kind of MRC task into following four layers. (i) *Embedding layer* that aims to obtain an embedding representation for each word in the input question and documents. This layer can also be used to obtain the basic semantic meaning of a word, but it could not well address the common issues of OOV words, polysemy phenomenon, and synonymy phenomenon in natural language. Some researchers integrate extra language models like *BERT* [Devlin et al. 2018] or *XLNet* [Yang et al. 2019a] into this layer, which can alleviate above issues but the cost is introducing too many parameters. The models with large amount of parameters require very large memory hardware, which may be unaffordable to many users. (ii) *Matching layer* that is used for mining the interactions between the input question and documents. It is often the core module in most existing MRC models and has been widely explored. Lots of attention based methods are proposed in this layer. For example, *BiDAF* [Seo et al. 2016] designs a context-to-query and query-to-context bi-directional attention method. Many other researchers, such as [Clark and Gardner 2018; Yu et al. 2018], also use a *BiDAF*-style attention method in this layer. Besides, [Cui et al. 2017] design an attention-over-attention method. [Wang et al. 2018c] design a multi-granularity hierarchical attention method. [Yan et al. 2019] use the self-attention method. (iii) *Model layer* that often uses *LSTM* or *CNN* based methods to capture the interactions among documents' words conditioned on the question features. (iv) *Prediction layer* that often uses the *pointer networks* to predict the probability of each position in the context being the start or end of an answer.

It should be noted that the emergence of *BERT* [Devlin et al. 2018] and lots of its variants (like *XLNet* [Yang et al. 2019a], *RoBERTa* [Liu et al. 2019], and *ALBERT* [Lan et al. 2020], etc.) greatly boost the benchmark performance of current MRC models due to their strong capacity for capturing the contextualized sentence-level language representations¹ [Zhang et al. 2021]. These language models simplify the building of an MRC model and lots of most recent MRC models [Banerjee et al. 2021; Chen and Wu 2020; Gong et al. 2020b; Guo et al. 2020a,b; Huang et al. 2020; Li et al. 2020b,a; Long et al. 2020; Luo et al. 2020; Zhang et al. 2020c; Zheng et al. 2020] only consist of a language model based encoder module and an MRC task specific decoder module. However, there is a fatal deficiency for these language models. First, except *XLNet*, *BERT* and its other variants (*ALBERT*, *RoBERTa*, etc.) are auto-encoding based models, which limits input size of 512 TOKENS [Gong et al. 2020b; Zemlyanskiy et al. 2021]. This restriction has no effect on most AI-related applications and most of single-document MRC tasks, but for a multi-document MRC dataset like DuReader or TriviaQA Web, this restriction will make most correct answers be excluded from the input documents even after a carefully designed data selection module. As for *XLNet*, it is an auto-regressive based model, and can handle long text theoretically. However, it is an uni-directional model which can make predictions based on forward information only, and can not use the backward information.

2.2 Multi-document MRC

For this kind of MRC task, researchers often design similar layers as in the single-document MRC task but integrate new techniques to make full use of the multi-document information. Initially,

¹In most pretrained language model based models, like the *BERT*-based models, a separated token *CLS* is often padded to the beginning of an input sentence, and its embedding representation is believed to contain the general information of the whole input sentence, and is often used as a representation of this sentence.

researchers use simple reading strategies. For example, [Wu et al. 2018] convert the multi-document data into the single-document format and then use single-document MRC models to find answers. For example, [Clark and Gardner 2018] first predict which paragraph to read and then apply models like *BiDAF* to pinpoint the answer within that paragraph. Obviously, these simple methods could not make full use of information contained in the multi-documents, thus researchers begin to design more sophisticated models to address the multi-document MRC task. For example, [Wang et al. 2018c] design three different modules in their model, which can find the answer boundary, model the answer content, and perform cross-passage answer verification respectively. [Yan et al. 2019] develop a novel deep cascade learning model that progressively evolves from the document-level and paragraph-level ranking of candidate texts to a more precise answer extraction. [Xu et al. 2019] propose a multitask learning model with a sample re-weighting scheme.

Recently, the models of imitating reading patterns used by human are achieving more and more research attention due to their competitive results on many benchmark MRC datasets. For example, [Sun et al. 2019] explicitly use three human's reading strategies in their MRC model, including: (1) back and forth reading, (2) highlighting, and (3) self-assessment. [Wang et al. 2018a] imitate human's following reading pattern: first scans through the whole passage; then with the question in mind, detects a rough answer span; finally, come back to the question and select a best answer. [Liu et al. 2018] design their MRC model by simulating human's multi-step reasoning pattern: human often re-read and re-digest given documents many times before a final answer is found. [Wang et al. 2018b] use an extract-then-select reading strategy. They further regard the candidate extraction as a latent variable and train the two-stage process jointly with reinforcement learning. [Peng et al. 2020] design their MRC model by simulating two ways of human thinking when answering questions, including reverse thinking and inertial thinking. [Zhang et al. 2021] imitate human's "*read + verify*" reading pattern: first to read through the full passage along with the question and grasp the general idea, then re-read the full text and verify the answer. Some other researchers [Clark and Gardner 2018; Hu et al. 2019; Wang et al. 2018c; Yan et al. 2019] also imitate human's "*read + verify*" reading pattern. Besides, there are other kinds of human reading patterns imitated, like the pattern of restoring a scene according to the text to understand the passage comprehensively [Tian et al. 2020], the pattern of human gaze during reading comprehension [Malmaud et al. 2020], the pattern of tactical comparing and reasoning over candidates while choosing the best answer [Chen et al. 2020], etc.

Here we classify all these existing models as a kind of *shallow understanding* based methods, since they pay more attention to these reading patterns' *superficial frameworks*, but ignore some important *understandings* hidden in these patterns.

3 METHODOLOGY

The framework of our model is shown in Figure 1. It mainly consists of three *understanding* modules that are designed to imitate human's three kinds of *understandings* respectively.

Given a question and some documents, the *Accurate Word Semantic Meaning Understanding* module will generate a vector representation for each word in these input texts. These vector representations are expected to contain the accurate semantic meaning of words when considering them in the overall context (including the question and the given documents). Then the *Interaction Understanding* module further mines the interactions between the question and its documents, and outputs a new vector representation for each word in the documents. In each of these vector representations, the question-aware features are integrated. It should be noted that the input of this module includes all the vector representations that correspond to the words in both the question and its documents, but the output of this module only includes the vector representations that correspond to the words in the documents. Next, taking these vector representations as input,

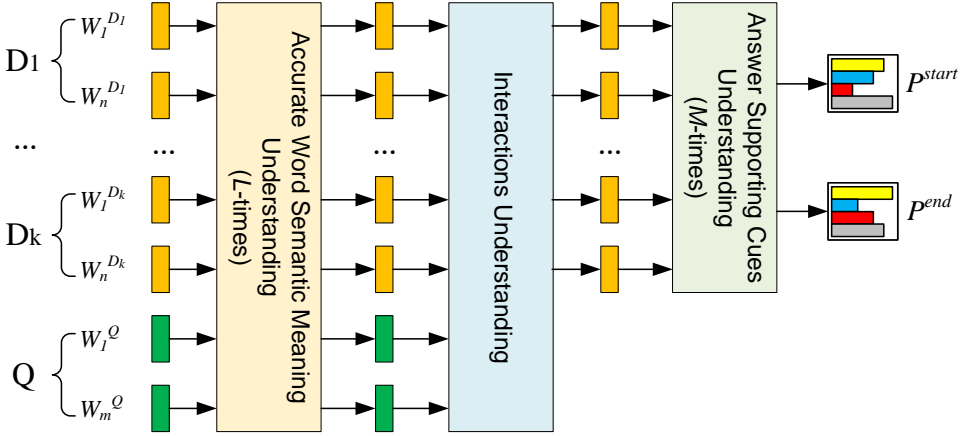


Fig. 1. The Framework of Our Model.

the *Answer Supporting Cue Understanding* module further mines the cue information which can indicate the possibility of a word in the documents being a context word in an answer. This module will output a refined vector representation for each word in the documents. Finally, based on the newest vector representations, the model computes the probabilities of each word being the start and end tokens of an answer. Based on these probabilities, an answer can be deduced.

3.1 Accurate Word Semantic Meaning Understanding

Accurately understanding the semantic meaning of words in the input question and documents is often the first and most basic step when human readers solve a reading comprehension task. Accordingly, the aim of this module is to fulfill this kind of understanding. The embedding layer in traditional MRC models can achieve this aim to some extent. However, these embedding based methods could not accurately understand a word's semantic meaning because they generate a semantic representation for each word by only taking limited context (often the text where the word occurs) into consideration, which makes the generated representations are not expressive enough especially when there are phenomena of OOV, polysemy and synonymy in the input text. Some researchers [Yang et al. 2019b,c; Zhang et al. 2020b] explore to integrate extra pretrained language models in the embedding layer, which can alleviate above issues but at the cost of introducing more parameters. [Dai et al. 2020] introduce a token-level dynamic reader to select important intermediate words according to boundary words, but they do not aim to understand the semantic meaning of words.

We notice that the input question and documents are often highly related to each other. Thus the input question and documents can be viewed as the context of each other, which means the information from one part is useful to understand the semantic meaning of words in another part. For example, in Table 1, when the word "Tanzania" in the question is an OOV word, it would be very difficult to *understand* its semantic meaning when only considering the context of the question itself. However, it is still possible to obtain the expected semantic meaning when placing it in the given documents. Inspired by this observation, we design a *coarse-to-accurate* method for *word semantic meaning understanding*. Specifically, we first use a word embedding based method to obtain a word's shallow semantic meaning, then refine this meaning by integrating context information. This is also in line with human readers' reading pattern that they often grasp the

literal meaning of a word firstly, and then verify this meaning by placing it into different contexts to obtain the accurate semantic meaning of this word in the given context.

Coarse Word Semantic Meaning Understanding Given a question and k documents, we adopt some widely used word embedding techniques² to generate $k + 1$ word representation sequences for them. We use $h^Q = (h_1^Q, h_2^Q, \dots, h_m^Q)$ and $h^{D_t} = (h_1^{D_t}, h_2^{D_t}, \dots, h_{n_t}^{D_t})$ to denote the sequences for the question Q and its t -th document D_t respectively, where m and n_t are the numbers of words in the question and the t -th document. Each item in these sequences can be viewed as a *coarse* semantic meaning for the corresponding word. Then these *coarse* semantic meaning will be refined by integrating context information to get the final *accurate* semantic meaning.

Accurate Word Semantic Meaning Understanding This step is expected to: (i) highlight the accurate semantic meaning of words in documents from the perspective of the question, and (ii) highlight the accurate semantic meaning of words in a question from the perspective of documents. Obviously, this expectation matches the principle of *attention* mechanism well. Thus, here we design a *cross attention* based method to obtain the accurate semantic meaning of words in the input question (or documents) by taking documents (or the question) as context. Specifically, the designed methods has following four steps.

Step 1: we first compute a cross-attention matrix A , each of its element $A_{i,j}$ indicates the relevance between the i -th word in D_t and the j -th word in Q . A is computed with Eq. (1).

$$A_{i,j} = (h_i^{D_t})^T \mathbf{W} h_j^Q + \mathbf{U}_l \odot h_i^{D_t} + \mathbf{U}_r \odot h_j^Q \quad (1)$$

where \mathbf{W} , \mathbf{U}_l and \mathbf{U}_r are trainable matrices, \odot denotes the inter production operation, and in all of this paper, the superscript T denotes a transpose operation.

Step 2: we assign an attention weight for each word in the input question and documents. And the attention weight of a word is computed with Eq. (2).

$$\begin{aligned} \alpha_i^{D_t} &= \text{softmax}(A_{i,:}) \\ \alpha_j^Q &= \text{softmax}(A_{:,j}) \end{aligned} \quad (2)$$

Step 3: based on the attention weights generated in previous step, we generate $\tilde{h}_i^{D_t}$ and \tilde{h}_j^Q , which are new representations for a word in a document D_t and a word in the question Q respectively. They are computed with Eq. (3).

$$\tilde{h}_i^{D_t} = h^Q \alpha_i^{D_t}, \quad \tilde{h}_j^Q = h^{D_t} \alpha_j^Q \quad (3)$$

Step 4: we perform a bi-directional GRU based fusion operation to further refine the results generated in previous step, as shown in Eq. (4) and Eq. (5), where $\mathbf{W}_f^{(\cdot)}$ are trainable matrices.

$$\begin{aligned} f_i^{D_t} &= [h_i^{D_t}; h_i^{D_t} - \tilde{h}_i^{D_t}; h_i^{D_t} \odot \tilde{h}_i^{D_t}]; \\ \tilde{f}_i^{D_t} &= \text{Relu}(\mathbf{W}_f^D f_i^{D_t} + b_f); \\ \bar{f}_i^{D_t} &= \text{BiGRU}(f_i^{D_t}, \bar{f}_{i-1}) \end{aligned} \quad (4)$$

$$\begin{aligned} f_j^Q &= [h_j^Q; h_j^Q - \tilde{h}_j^Q; h_j^Q \odot \tilde{h}_j^Q]; \\ \tilde{f}_j^Q &= \text{Relu}(\mathbf{W}_f^Q f_j^Q + b_f); \\ \bar{f}_j^Q &= \text{BiGRU}(\tilde{f}_j^Q, \bar{f}_{j-1}) \end{aligned} \quad (5)$$

²Here we use the GloVe word embeddings [Pennington et al. 2014], the character embeddings that are generated by a common CNN model, and the highway network. All of them are widely used in existing MRC models.

The resulted $\bar{f}_i^{D_t}$ and \bar{f}_j^Q denote the new representations for the i -th word in D_t and the j -th word in Q , each of them corresponds to a refined semantic meaning of a word.

As shown in Figure 1, the above four steps will be iterated L times to obtain the final *accurate* semantic meaning for each word. This repeated manner has been proven to be effective [Liu et al. 2018] for an MRC task. It should be noted that the matrices or vectors used in above equations are different for each iteration. For example, there will be L different W in Eq. (1). Here for simplicity, we do not make distinction in the equation descriptions.

Finally, this *accurate word semantic meaning understanding* module outputs a new embedding representation for each word in the question and the given documents. We denote the final word representation sequences for Q and D_t as $\bar{f}^Q = (\bar{f}_1^Q, \bar{f}_2^Q, \dots, \bar{f}_m^Q)$ and $\bar{f}^{D_t} = (\bar{f}_1^{D_t}, \bar{f}_2^{D_t}, \dots, \bar{f}_{n_t}^{D_t})$ respectively. And each item in these sequences can be viewed as the final *accurate* semantic meaning for the corresponding word.

3.2 Interaction Understanding

This module aims to find some important cues from the given documents that are helpful for locating an answer. The difficulty for achieving this goal is how to accurately understand the interactions between the input question and documents. Different from the *word semantic meaning understanding* that mainly focuses on the word-level understanding, this module will focus on the document-level (or paragraph-level if we view each document and the question as a paragraph) semantic meaning understanding. We notice that when human readers solve this problem, they often first analyze the interactions between the input question and documents, then keep the question in mind and re-read the documents to find the answer. Inspired by this, we design a *two-step interaction understanding* method that is similar to human's above reading strategy. Specifically, it first analyzes the interactions between the input question and documents, then integrates the question features into the representations of words in documents to form a question-aware representation for each word in documents.

Step 1: in this step, it is a natural way to design a bi-directional attention based method due to the following two reasons. First, interactions are always bi-directional. Second, understanding interactions is to find which words are more helpful from the perspective of finding an answer, which is in line with the principle of attention mechanism.

The attention method used in *BiDAF* [Seo et al. 2016] has been proven to be a very powerful method for understanding the interactions between the input question and documents and is widely used by lots of existing MRC models [Clark and Gardner 2018; Yu et al. 2018]. Thus in this step, we use the same attention method with *BiDAF*. We omit the description of this attention computation process and readers can find the detail information in the original paper. Here we directly use $\{h^{D_t, 2Q}\} \in R^{n_t \times d}$ and $\{h^{Q, 2D_t}\} \in R^{m \times d}$ to denote the resulted document-to-question and question-to-document attended vectors, which are outputted by the *BiDAF* based method.

Step 2: in this step, we also use a *BiDAF*-like fusion method to combine the attention vectors and the embeddings obtained in previous *word semantic meaning understanding* module together to yield a document representation sequence G , each of its items $g^{D_t} \in R^{n_t \times d}$ denotes a new representation for a document D_t where the question-aware information is integrated. But what's different with *BiDAF* is that here we use a *BiGRU* based fusion function other than a simple concatenation operation. Specifically, g^{D_t} is computed with Eq. (6).

$$g_i^{D_t} = \text{BiGRU}(g_{i-1}^{D_t}, [\bar{f}_i^{D_t}; h_i^{D, 2Q}; \bar{f}_i^{D_t} \odot h_i^{Q, 2D}; \bar{f}_i^{D_t} \odot h_i^{D, 2Q}]) \quad (6)$$

It should be noted that as shown in Figure 1, we do not perform a repeated operation in this module. This is because that the input of this module contains word representations of both the input question and documents, but the output only contains the word representations of documents.

Thus the number of input tokens is different from the number of output tokens. Of course, a linear transformation can be used to map the output of this module to the same size as the input. But this would be lack of a reasonable explanation: in the original input, each representation correlates with a real word either in the question or in its documents, but it is very difficult to ask the transformed results still can be semantically correlated with these input words. In fact, our subsequent experiments show that repeating this module with a linear transformation operation is much harmful to the performance of our model.

3.3 Answer Supporting Cue Understanding

In the multi-document MRC, every document is expected to contain the answer or some information that is highly related to the answer, thus the semantic meaning of different documents would be highly related to each other. Accordingly, if an answer candidate in a document is the correct answer, it would be highly possible to achieve extra supporting cues³ from other documents. Besides, the word representations generated by previous *interaction understanding* module are question-aware, so a correct answer would also be highly possible to achieve extra supporting cues from words in the same document. Based on these analyses, we design an *intra-document* and *inter-document* self-attention based method to collect these supporting cues.

Intra-document Answer Supporting Cue Understanding is a self-attention based method that is designed to highlight the answer's content words from the perspective of other words in the same document where these content words appear. In other words, this module is expected to highlight some words that are regarded as answer words by most words. Specifically, it generates $f^{D_t} \in R^{n_t \times d}$, a new word representation sequence for each document D_t , as shown in Eq. (7).

$$f_i^{D_t} = BiGRU(f_{i-1}^{D_t}, [g_i^{D_t}, w_i]) \quad (7)$$

where $g_i^{D_t}$ is the representation of the i -th token in the document D_t and is generated by previous *interaction understanding* module, w_i is the attention value between $g_i^{D_t}$ and g^{D_t} , and is computed with a widely used attention computation method [Bahdanau et al. 2015; Wang et al. 2017] as shown in Eq. (8), where v is a trainable vector, \mathbf{W}^{D_t} and \mathbf{V}^{D_t} are trainable matrices.

$$\begin{aligned} s_i^j &= v^T \tanh(\mathbf{W}^{D_t} g_i^{D_t} + \mathbf{V}^{D_t} g_j^{D_t}), \\ \alpha_i^j &= \exp(s_i^j) / \sum_{j=1}^{n_t} \exp(s_i^j), \\ w_i &= \sum_{j=1}^{n_t} \alpha_i^j g_j^{D_t} \end{aligned} \quad (8)$$

Inter-document Answer Supporting Cue Understanding is design to highlight the answer's content words from the perspective of other documents. In other words, this module is expected to highlight some words that are regarded as answer words by most documents. Specifically, for each document, we first concatenate all its words' representations obtained in previous *intra-document supporting cue understanding* step together to form a new representation for this document. Accordingly, we will obtain a new document representation sequence $P = \{f_1^{D_1}, f_2^{D_1}, \dots, f_1^{D_k}, \dots, f_{n_t}^{D_k}\}$, and each item in this sequence corresponds to the representation of a document. Then the inter-document self-attention is performed on P to generate $F_p = \{f_p^1, f_p^2, \dots, f_p^L\}$, where $L = \sum_{i=1}^k n_i$. Each of its item f_p^i corresponds to the representation of a word in the concatenated document, and

³Here we define the answer supporting cues as a kind of information that is very helpful for locating an answer.

is computed with the method show in Eq. (9) and (10).

$$f_p^i = \text{BiGRU}(f_p^{i-1}, [P_i; \beta_i]) \quad (9)$$

$$s_i^j = \gamma^T \tanh(\mathbf{W}_f P_i + \mathbf{V}_f P_j),$$

$$\mu_i^j = \exp(s_i^j) / \sum_{j=1}^L \exp(s_i^j), \quad (10)$$

$$\beta_i = \sum_{j=1}^L \mu_i^j P_j$$

where \mathbf{W}_f and \mathbf{V}_f are trainable matrices, and γ is a trainable vector.

As shown in Figure 1, the *answer supporting cue understanding* module will be repeated M times so that more accurate supporting cues are highlighted. Finally, we still denote the output of this module as F_p , each item of which corresponds to a word representation where different kinds of *understanding* information is integrated.

3.4 Answer Prediction

We use a pointer networks based method that is similar to the ones in BiDAF [Seo et al. 2016] and Match-LSTM [Wang and Jiang 2017] to predict the probability of each word in F_p being the start or the end of an answer span. The pointer networks [Vinyals et al. 2015] produce only the start token and the end token of an answer, and then all the tokens between these two tokens in the original passage are considered to be the correct answer. Specifically, the probability distributions of the start and end indexes over tokens of all documents are computed with Eq.(11).

$$\begin{aligned} \mathbf{P}^s &= \beta_s^T \tanh(\mathbf{W}_p^s F_p + \mathbf{W}_g^s G) \\ \mathbf{P}^e &= \beta_t^T \tanh(\mathbf{W}_p^e F_p + \mathbf{W}_g^e G) \end{aligned} \quad (11)$$

where $\mathbf{W}_p^{(\cdot)}$ and $\mathbf{W}_g^{(\cdot)}$ are trainable matrices, $\beta_{(\cdot)}$ are trainable vectors, and G is the output of the previous *interaction understanding* module. Note that G has the same number of tokens as F_p .

Finally, we define the loss function as the negative sum of the log probabilities of the predicted distributions indexed by the true start and end indices over all samples, as shown in Eq. (12).

$$\text{Loss} = - \sum_i^N [\log(p_{y_i^s}^b) + \log(p_{y_i^e}^e)] \quad (12)$$

where y_i^b and y_i^e are the true start and end index of the i -th sample respectively.

At the inference time, an answer candidate A'_i (we denote its start and end indices as x and y respectively) is chosen with the maximum value of $a_x^b a_y^e$ under a constraint that $x \leq y$.

4 EXPERIMENTS

4.1 Datasets and Experimental Settings

We evaluate our model on TriviaQA Web [Joshi et al. 2017] and DuReader [He et al. 2018], two large-scale multi-document MRC benchmark datasets.

TriviaQA is an English MRC dataset containing over 650K question-answer-evidence triples. It includes 95K question-answer pairs authored by trivia enthusiasts and independently gathered evidence documents, six per question on average, which are generated from either Wikipedia or Web search. **Note** there are two separated datasets in TriviaQA: one is TriviaQA Wiki which is for the single-document MRC, and the other is TriviaQA Web which is for the multi-document

MRC. In TriviaQA Web, besides the full development and test set, a verified subset for each is also provided.

DuReader is a Chinese multi-document MRC dataset, which is designed to address real-world MRC. It has three advantages over previous MRC datasets. First, all of its questions and documents are based on Baidu Search and Baidu Zhidao, and the answers are manually generated. Second, it provides rich annotations for more question types. Third, it is a large MRC dataset that contains 200K questions, 420K answers and 1M documents.

Implementation Details In experiments, the dimension of word embeddings and the hidden layer in the *BiGRU* unit are set to 300 and 150 respectively. During training, word embeddings are not updated and the batch size is set to 16. Adam optimizer is used and the learning rate is set to 0.001. Training epoch is set to 2. On DuReader test set leader-board, ROUGH-L and BLEU4 are used as evaluation metrics. On TriviaQA Web test set leader-board, EM and F1 are used as evaluation metrics. In experiments, the ensemble model is obtained by averaging 4 single models' prediction probabilities. DuReader provides *free-form* reference answers that not all can be found in the input documents. So for each question, as the method used in [Wang et al. 2018a], we choose the span that achieves the highest ROUGE-L score with its reference answers as the golden span for training.

During training, we also design a simple string matching based data preprocessing module to filter out some irrelevant sentences from each question's given documents. Specifically, we compute a *cosine* similarity between each sentence of input documents and the ANSWER⁴. Only the sentences whose similarities are higher than a predefined threshold would be left for model training. During testing, answers are not available, we select the sentences by computing the *cosine* similarity between sentences in the input documents and the questions.

Baselines Following strong state-of-the-art models are taken as baselines: BiDAF [Seo et al. 2016], Smartnet [Chen et al. 2017], Fast[Wu et al. 2018], Simple [Clark and Gardner 2018], QANet [Yu et al. 2018], Cascade [Yan et al. 2019], Match-LSTM [Wang and Jiang 2017], R-Net [Wang et al. 2017], PR+BiDAF [Wang et al. 2018c], CrossPassage [Wang et al. 2018a], and BCTN [Peng et al. 2020]. All of them are the best multi-document MRC models that can be found so far⁵. Here except the results of BCTN, all the results of other baselines are directly copied from [Yan et al. 2019]. Besides, we also report the results of two popular pretrained language models: one is *RoBERTa* [Liu et al. 2019], and the other is *XLNet* [Yang et al. 2019a]. Both of them are recent variants of *BERT* and are reported to be superior to *BERT* or other kinds of language models like *Elmo* [Peters et al. 2018] on a lot of AI-related tasks. As a strong variant of *BERT*, *RoBERTa* can use the *sliding window* based method to handle the text that is longer than 512 tokens⁶. Besides, it should be noted that both language models have two versions: base and large. Here we report their results of both versions.

4.2 Main Experimental Results

Because we do not have high performance (like large-memory or fast speed) GPU servers, we first use 30,000 training samples and 6,000 testing samples of DuReader to quickly find the proper settings of L and M . Then we report all the other experiments based on these two fixed parameters. The results are shown in Table 2, from which we can see that ROUGE-L and BLEU4 do not increase synchronously. This is mainly because the provided answers in DuReader is *free-form*, and we need to convert these reference answers into spans to fit in with the span extraction based methods. During this process, ROUGE-L is used as the guide metric. So it is possible to make ROUGE-L

⁴In our in-house experiments, we find that using *answer* achieves better experimental results than using "question".

⁵It should be noted that there are some models that appear on the test set leader-boards of some MRC datasets, but we could not find their corresponding papers either in conferences, journals, or on arXiv.

⁶The used *RoBERTa* model is implemented by a transformer code base, which can be found at following website: <https://github.com/huggingface/transformers/>.

Table 2. Effect of repeated numbers on DuReader.

	ROUGE-L/BLEU4			
	$M=1$	$M=2$	$M=3$	$M=4$
$L=1$	53.12/50.74	54.23 /50.52	52.41/50.39	50.03/47.87
$L=2$	52.32/48.29	52.83/49.03	51.44/45.27	47.58/48.69
$L=3$	50.26/49.39	51.02/ 53.28	46.17/48.24	45.66/47.19
$L=4$	48.13/48.28	48.69/50.81	47.32/49.29	44.33/44.75

Table 3. Main results on DuReader. * indicates the results are generated by us. † indicates that the results are copied from [Peng et al. 2020] directly.

	ROUGE-L	BLEU4
MatchLSTM [Wang and Jiang 2017]	39	31.8
BiDAF [Seo et al. 2016]	39.2	31.9
R-Net [Wang et al. 2017]	47.71	44.88
PR+BiDAF [Wang et al. 2018c]	41.81	37.55
CrossPassage [Wang et al. 2018a]	44.18	40.97
CascadeModel [Yan et al. 2019]	50.71	49.39
<i>XLNet-Base</i> [Yang et al. 2019a]	57.36*	49.21*
<i>XLNet-Large</i> [Yang et al. 2019a]	61.05*	54.38*
RoBERTa-Base [Liu et al. 2019]	54.18†	38.85†
RoBERTa-Large [Liu et al. 2019]	59.12†	44.53†
BCTN-Base [Peng et al. 2020]	58.04	43.19
BCTN-Large [Peng et al. 2020]	59.12	44.53
<i>OurModel (Single)</i>	62.19	56.34
<i>OurModel (Ensemble)</i>	63.36	57.91

Table 4. Main results on TriviaQA Web. * indicates the results are generated by us.

Model	Full	Verified
	EM/F1	EM/F1
BiDAF [Seo et al. 2016]	40.74/47.05	49.54/55.80
Smarnet [Chen et al. 2017]	40.87/47.09	51.11/55.98
Fast [Wu et al. 2018]	47.77/54.33	57.35/62.23
Simple [Clark and Gardner 2018]	66.37/71.32	79.97/83.70
QANet [Yu et al. 2018]	51.1/56.6	53.3/59.2
Cascade [Yan et al. 2019]	68.65/73.07	82.44/85.35
RoBERTa-Base [Liu et al. 2019]	64.97*/70.89*	78.41*/83.03*
RoBERTa-Large [Liu et al. 2019]	66.65*/72.39*	79.84*/84.49*
<i>XLNet-Base</i> [Yang et al. 2019a]	63.92*/67.42*	77.39*/79.57*
<i>XLNet-Large</i> [Yang et al. 2019a]	65.64*/69.40*	79.58*/82.08*
<i>OurModel (Single)</i>	68.72/73.13	82.70/85.35
<i>OurModel (Ensemble)</i>	69.64/73.80	83.36/85.66

and BLEU4 reach their peaks under different conditions, which will then cause the mentioned phenomenon. In fact, this phenomenon is also common on other *free-form* MRC datasets like MS

MARCO [Nguyen et al. 2016]. Since DuReader test set leader-board takes ROUGE-L as the main evaluation metric, all the experiments are reported under the settings of $L = 1$ and $M = 2$ where the model achieves the highest ROUGE-L score.

The main experimental results are summarized in Table 3 and 4. From these results we can see that our model is very effective: on both datasets and under all evaluation metrics, it consistently outperforms all the compared state-of-the-art baselines.

Furthermore, we can see that on both datasets, our model achieves much better results than both *RoBERTa* and *XLNet*. We argue this is mainly due to following reasons. For *RoBERTa*, its *sliding window* mechanism can alleviate the issue of handling long documents. This mechanism slices a document into multiple segments, and each segment will be individually encoded by the encoder, finally all the encoded results of these segments are merged. This will lead to following fatal deficiency. In this mechanism, each slice is encoded separately, which will lose much of important correlation information among documents. Especially when the answer length exceeds 512 tokens, this mechanism will make the semantic meaning of different slices incomplete, which is very harmful for finding some important cues from either the intra-document level or the inter-document level. This deficiency will harm the performance greatly. As for *XLNet*, although it can handle long text due to its auto-regressive mechanism, its uni-directional processing property still makes it can not make full use of the given documents due to the lose of backward information.

4.3 Ablation Experiments

To demonstrate the contributions of different components in our model, we conduct ablation experiments and the results are shown in Table 5.

Effectiveness of Accurate Word Semantic Meaning Understanding From Table 5 we can see that when the *accurate word semantic meaning understanding* module is removed, both ROUGE-L and BLEU4 drop sharply on DuReader. Similar results can be seen on TriviaQA Web. These results show that it is helpful for accurately understanding the semantic meaning of words in the input question and documents.

To further evaluate the effectiveness of the *accurate word semantic meaning understanding* module, we replace it with *XLNet*. In other words, we use *XLNet* on top of the common word embedding layer since this practice is taken by lots of existing models. New experiments are shown in Table 6 (Here the large version of *XLNet* is used.). We can see that our designed *word semantic meaning understanding module* works better. We think this is mainly due to following two reasons. First, as analyzed above, *XLNet* can not make full use of the given documents due to the lose of backward information. Second, *XLNet* is a pretrained model, some words' semantic meaning generated by it may not well match the true scenario in an MRC dataset. One may argue that the parameters in *XLNet* can be re-trained on a specific application scenario. But training large-scale language models is often very time-consuming and the required hardwares (such as GPU servers) are far beyond what we can afford.

Besides, we can see that when *XLNet* is integrated into our model, it achieves much better results on both datasets than its original version. These results indicate that our proposed model has a general framework that can be used to further boost the performance of existing MRC models.

Effectiveness of Answer Supporting Cue Understanding We can see that when the whole *answer supporting cue understanding* module is removed, the performance drops sharply. But when either the *intra-document* or the *inter-document answer supporting cue understanding* module used, the model achieves competitive results. Besides, the performance drops more when the *inter-document answer supporting cue understanding* module is removed, which shows the supporting cues from other documents play more roles than that of from a document itself. This is just like a

Table 5. Ablation experiments on TriviaQA Web (*upper part*) and DuReader (*bottom part*).

<i>Model</i>	Full	Verified
	EM/F1	EM/F1
<i>OurModel(Single)</i>	68.72/73.13	82.70/85.35
<i>-CrossAttU</i>	67.19/71.01	77.86/82.45
<i>-InteractionU</i>	64.21/68.68	75.47/78.17
<i>-SupportingCueU</i>	60.12/62.97	71.32/72.45
<i>-Intra-DocSelfAtt</i>	68.38/72.62	81.92/84.69
<i>-Inter-DocSelfAtt</i>	66.87/71.22	79.68/83.87
<i>-DataPreprocessing</i>	66.43/71.35	79.97/83.70
Model	ROUGE-L	BLEU4
<i>OurModel(Single)</i>	62.19	56.34
<i>-CrossAttU</i>	61.37	55.04
<i>-InteractionU</i>	58.78	54.17
<i>-SupportingCueU</i>	54.30	49.21
<i>-Intra-DocSelfAttn</i>	60.25	55.43
<i>-Inter-DocSelfAtt</i>	58.78	54.39
<i>-DataPreprocessing</i>	60.04	54.46

Table 6. Comparisons of with/without *XLNet* on TriviaQA Web (*upper part*) and DuReader (*bottom part*).

<i>Model</i>	Full	Verified
	EM/F1	EM/F1
<i>OurModel(Single)</i>	68.72/73.13	82.70/85.35
<i>OurModel+XLNet</i>	66.09/69.78	79.71/81.88
	ROUGE-L	BLEU4
<i>OurModel(Single)</i>	62.19	56.34
<i>OurModel+XLNet</i>	61.50	54.79

voting process: the more documents provide supporting cues, the more likely an answer candidate be the correct answer.

Effectiveness of Interaction Understanding From Table 5 we can see that the *interaction understanding* modules are important.

In fact, our model is adaptable to different choices other than *BiDAF* in the *Interaction Understanding* module. To evaluate this adaptability, we use the interaction methods in several other MRC models to replace *BiDAF*, and the results are shown in Table 7. We can see that all these models have similar contributions as *BiDAF*. Furthermore, from Table 7 we can see that when a model is integrated into the framework of our model, it always achieves much better results than its original version. Taking Match-LSTM as example, when it is used in our model, its results on both datasets are far higher than those of its original version. These results confirm again that the proposed model has a general framework and can be used to further boost the performance of existing MRC models.

Besides, we also conduct experiments that perform a repeated operation in this *interaction understanding* module by a simple linear transformation operation. The results are shown in Table 8 (the results are obtained under our *single* version model). We can see that there is a significant performance drop when this module begins to repeat ($N = 1$). Then, as the the repeated number

Table 7. Comparisons of using different *interaction understanding* methods on TriviaQA Web (*upper part*) and DuReader (*bottom part*).

<i>Model</i>	Full	Verified
	EM/F1	EM/F1
+BiDAF(<i>OurModel</i>)	68.72/73.13	82.70/85.35
+MatchLSTM[Wang and Jiang 2017]	66.54/72.28	80.33/84.12
+AOA[Cui et al. 2017]	66.77/72.45	81.31/84.89
+RNet[Wang et al. 2017]	67.94/72.42	81.22/84.97
	ROUGE-L	BLEU4
+BiDAF(<i>OurModel</i>)	62.19	56.34
+MatchLstm[Wang and Jiang 2017]	61.42	54.73
+AOA [Cui et al. 2017]	61.05	54.38
+RNet[Wang et al. 2017]	61.31	55.74

Table 8. Effect of repeated numbers (N) for the *interaction understanding* module on DuReader and TriviaQA.

DuReader (ROUGE-L/BLEU4)			
$N=0$	$N=1$	$N=2$	$N=3$
62.19/56.34	60.37/55.92	58.26/54.12	53.87/48.53
TriviaQA Full (EM/F1)			
$N=0$	$N=1$	$N=2$	$N=3$
68.72/73.13	66.51/70.34	63.14/66.25	59.67/64.28
TriviaQA Verified (EM/F1)			
$N=0$	$N=1$	$N=2$	$N=3$
82.70/85.35	80.04/82.14	77.57/72.69	73.09/68.24

Table 9. Comparisons of parameter number (*millions*) on TriviaQA Web and DuReader.

	TriviaQA	DuReader
MatchLSTM[Wang and Jiang 2017]	≈128	≈93
BiDAF[Seo et al. 2016]	≈113	≈84
XLNet [Yang et al. 2019a]	≈146	≈123
<i>OurModel</i> +XLNet	≈243	≈212
<i>OurModel</i> (Single)	≈126	≈92

increases, the performance of our model drops accordingly. Especially, when $N = 3$, the performance of our model is even worse than that of removing the whole *interaction understanding* module. We take the sample shown in Table 1 as a specific case, and use a simple *inner-product* based method to compute the similarity between the original representation of the word “Tanzania” and its transformed representation. The results show that the similarities between these two representation become lower and lower as the repeated number increases. Such results confirms our previous analyses that: in the original input, each representation correlates with a real word either in the question or in its documents, but it is very difficult to ask the transformed results still can be semantically correlated with these input words. Thus there is a risk that after several repeated operations, the semantic meaning of the transformed results are far and far away from those of the input words, which would be much harmful to the performance of our model.

Table 10. Error Analyses. *Q* refers to *question*.

Error Types	Proportion(%)		Examples
	DuReader	TriviaQA Web	
incomplete	21.7	6.14	Q: A Long Island Iced Tea is a cocktail based on vodka, gin, tequila, and which other spirit? Golden answer: light rum Predicted answer: rum
redundant	35.5	23.68	Q: What does a costermonger sell? Golden answer: fruit Predicted answer: fruit and vegetables
unanswerable	0.3	10.53	Q: "Which US president was behind ""The Indian Removal Act"" of 1830, which paved the way for the reluctant and often forcible emigration of tens of thousands of American Indians to the West?" Golden answer: null Predicted answer: President Monroe
others	42.5	59.65	Q: Romaine & Butterhead are types of what? Golden answer: iceberg lettuce Predicted answer: lettuces

4.4 Parameter Efficiency

We quantitatively compare the parameter numbers of several models whose source codes are available. All the models are trained on a TitanRTX 8000 GPU server (*XLNet* requires so large memory that it couldn't be trained on a server like TitanXP) with the configurations that lead to the best performance we achieved. The comparison results are shown in Table 9. We can see that our model has less parameters than most of the compared models. When taking the performance into consideration, we can conclude that our model is more parameter efficient: it achieves better results with fewer parameters.

Here we do not compare the run time of different models because it is difficult to provide a fair evaluation environment: coding tricks, hyper-parameter settings (like *batch-size*, *learning rate*, etc), parallelization, lot of non-model factors affect the run time.

4.5 Error Analyses

Here we make some error analyses. Specifically, on DuReader, we randomly select 2,000 poorest ROUGE-L results generated by our model as error samples. And on TriviaQA Web, we take all the results whose EM values are wrong on the development set as error samples. Then we try to classify these error samples into different groups according to their error types, and the results are shown in Table 10, in which all the listed examples are taken from TriviaQA Web. For clarity, we omit the given documents of each example since these documents on either dataset are very long.

Generally, there are following three main kinds of errors on both datasets. (i) *incomplete*, which means that only partial of a predicted answer matches the corresponding golden answer. (ii) *redundant*, which means that a golden answer is a word subset of the predicted answer. (iii) *unanswerable*, which means that the question is unanswerable (an ideal model should identify these unanswerable questions and refuse to give an answer for it), but the model outputs an answer.

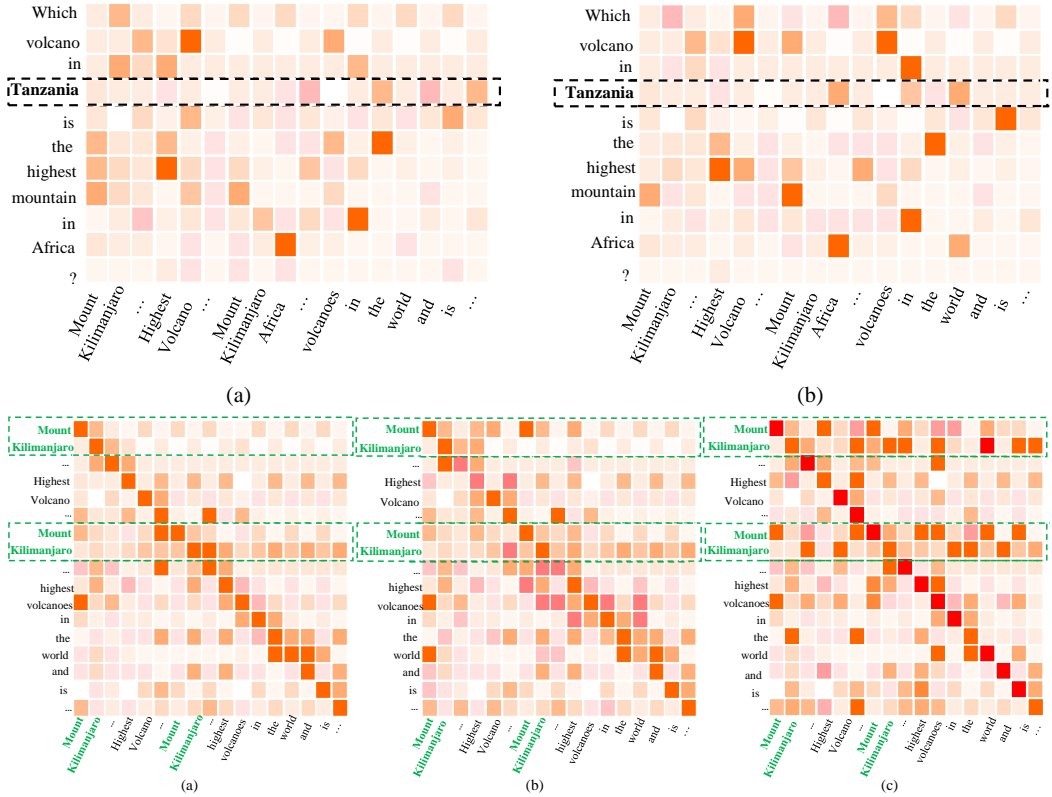


Fig. 2. **Upper:** comparisons of the *cosine* similarities between words’ embeddings of before (*subfigure a*) and after (*subfigure b*) using the proposed “*Accurate Word Semantic Meaning Understanding*” module. **Bottom:** comparisons of the attention weights when using “*Intra-document Answer Supporting Cue Understanding*” (*subfigure a*), “*Inter-document Answer Supporting Cue Understanding*” (*subfigure b*), and both (*subfigure c*).

On DuReader, both the *incomplete* and the *redundant* kind of errors account for a large proportion of all the errors. And the *other* kinds of errors includes *partial matching* errors, *yes/no* errors, etc. On TriviaQA Web, the *redundant* kind of errors account for a large proportion of all the errors, followed by the *unanswerable* kind of errors. The *unanswerable* kind of errors account for significantly larger proportion on TriviaQA Web than on DuReader because there are far less *unanswerable* kind of question on DuReader than that on TriviaQA Web. The *other* kind of errors on TriviaQA Web include errors like the *named entity recognition* errors, *singular and plural* errors, *partial matching errors*, etc. After detailed analyses of these errors we find that in most cases, the locations of the predicted answers are very close to the golden answers. In fact, these errors could be corrected only when a model do *understand* the main semantic meaning of the input text, which further indicates the reasonability of our research line.

4.6 Case Study

Taking the question and documents illustrated in Table 1 as an example, here we use Figure 2 to further demonstrate the effectiveness of the proposed two *understanding* modules. From the upper part of Figure 2 we can see that when the proposed “*Accurate Word Semantic Meaning Understanding*” module used, “*Tanzania*” (in the question) obtains higher similarities with words

like “Africa” and “world”, which highlights its accurate semantic meaning greatly. From the bottom part of Figure 2 we can see that the answer “Mount Kilimanjaro” achieves different attention weights when using different *answer supporting cue understanding* sub-module. When all the modules used, it achieves the highest attention weight, which increases the probability of it being the answer.

4.7 Discussions

Before this submission, the best results achieved by our model on TriviaQA Web and DuReader test set leader-boards were No.1⁷ and No.3⁸ respectively⁹. One can notice that currently, most top models on different MRC leaderboards (like SQuAD¹⁰, HotPotQA¹¹, CoQA¹², MS MARCO¹³, etc.) depend heavily on large scale pretrained language models like *BERT* (or its variants). However, these language models based MRC models have two fatal deficiencies.

First, they can only be run on high-cost hardware environments since the language models have so large amount of parameters that much large GPU memories are often required to load these parameters. This will bring heavy burdens on most researchers since the cost of building such environments are very high. Accordingly, this will prohibit these models to be used on some real-time or online scenarios.

Second, they can only be used on the scenarios where the maximum length of the input text is within a specific threshold since most existing language models like *BERT* (including most of its variants) have a length restriction on the input text. This condition is not always met, especially for some languages like Chinese where the corresponding MRC task usually involves very long text. One may argue that this deficiency can be addressed by re-training a new language model. However, re-training such a new large scale language model without length restriction is far beyond the affordability of most researchers due to the high hardware requirement and the high time cost.

Both deficiencies prohibit the adaptability of the language model based MRC models. On the contrary, our model is a simple and effective MRC model, and it has following two overwhelming advantages compared with the language model based MRC models.

First, our model uses simple technologies like GRU but achieve very competitive results, which means it can be very easily reproduced by other researchers.

Second, [Wang et al. 2020] have pointed out in their work that for all systems that use some pretrained language models like *BERT*, the language model is usually the most time-consuming part and takes up the most of model parameters. In contrast, our model does not use any pretrained language models, thus compared with the models that use per-trained language models, our model usually has a smaller parameter size and faster inference speed, which means it can well fit in with some online or real-time applications without the requirements of high-performance hardware.

In a word, our model shows that by well *understanding* the semantic meaning of the input text, the state-of-the-art performance still can be achieved even without using sophisticated technologies, high-cost hardware, and large scale language models.

5 CONCLUSIONS

In this paper, we propose a simple but effective *deep understanding* based multi-document MRC model. It uses neither any sophisticated technologies nor any pretrained language models. We

⁷<https://competitions.codalab.org/competitions/17208#results>, tab the *Web* button.

⁸<https://ai.baidu.com/broad/leaderboard?dataset=dureader&task=Main>

⁹Now it ranks No.2 and No.7 on these two test set leader-boards respectively.

¹⁰<https://rajpurkar.github.io/SQuAD-explorer/>

¹¹<https://hotpotqa.github.io/>

¹²<https://stanfordnlp.github.io/coqa/>

¹³<https://microsoft.github.io/msmarco/>

evaluate our model on DuReader and TriviaQA Web, two widely used benchmark multi-document MRC datasets. Experiments show that our model achieves very competitive results on both datasets.

The main novelties of our work are as follows. First, our model has a general framework that consists of three understanding modules that imitate human’s three kinds of understandings during reading comprehension. Second, the designed *accurate word semantic meaning understanding* module can well *understand* a word’s semantic meaning. It even plays a better role than an extra language model like *XLNet* but with far less parameters. This is very important for MRC’s application to the online or real-time environments. Third, the designed *answer supporting cue understanding* module is effective, and it can increase the probability of finding answers.

ACKNOWLEDGMENTS

This work is supported by the National Natural Science Foundation of China (No.61572120) and the Fundamental Research Funds for the Central Universities (No.N181602013).

REFERENCES

- Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural Machine Translation by Jointly Learning to Align and Translate. In *ICLR 2015 : International Conference on Learning Representations 2015*.
- Pratyay Banerjee, Tejas Gokhale, and Chitta Baral. 2021. Self-Supervised Test-Time Learning for Reading Comprehension. *arXiv preprint arXiv:2103.11263* (2021).
- Wuya Chen, Xiaojun Quan, Chunyu Kit, Zhengcheng Min, and Jiahai Wang. 2020. Multi-choice Relational Reasoning for Machine Reading Comprehension. In *Proceedings of the 28th International Conference on Computational Linguistics*. 6448–6458.
- Zheng Chen and Kangjian Wu. 2020. ForceReader: a BERT-based Interactive Machine Reading Comprehension Model with Attention Separation. In *Proceedings of the 28th International Conference on Computational Linguistics*. 2676–2686.
- Zheqian Chen, Rongqin Yang, Bin Cao, Zhou Zhao, Deng Cai, and Xiaofei He. 2017. Smarnet: Teaching Machines to Read and Comprehend Like Human. *arXiv preprint arXiv:1710.02772* (2017).
- Christopher Clark and Matt Gardner. 2018. Simple and Effective Multi-Paragraph Reading Comprehension. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Vol. 1. 845–855.
- Yiming Cui, Zhipeng Chen, Si Wei, Shijin Wang, Ting Liu, and Guoping Hu. 2017. Attention-over-Attention Neural Networks for Reading Comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Vol. 1. 593–602.
- Yilin Dai, Qian Ji, Gongshen Liu, and Bo Su. 2020. Multi-paragraph Reading Comprehension with Token-level Dynamic Reader and Hybrid Verifier. In *2020 International Joint Conference on Neural Networks (IJCNN)*. 1–8.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina N. Toutanova. 2018. 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: Human Language Technologies, Volume 1 (Long and Short Papers)*. 4171–4186.
- Chen Gong, Zhenghua Li, Qingrong Xia, Wenliang Chen, and Min Zhang. 2020a. Hierarchical LSTM with char-subword-tree-structure representation for Chinese named entity recognition. *Science in China Series F: Information Sciences* 63, 10 (2020), 202102.
- Hongyu Gong, Yelong Shen, Dian Yu, Jianshu Chen, and Dong Yu. 2020b. Recurrent Chunking Mechanisms for Long-Text Machine Reading Comprehension.. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 6751–6761.
- Shaoru Guo, Yong Guan, Ru Li, Xiaoli Li, and Hongye Tan. 2020a. Incorporating Syntax and Frame Semantics in Neural Network for Machine Reading Comprehension. In *Proceedings of the 28th International Conference on Computational Linguistics*. 2635–2641.
- Shaoru Guo, Ru Li, Hongye Tan, Xiaoli Li, Yong Guan, Hongyan Zhao, and Yueping Zhang. 2020b. A Frame-based Sentence Representation for Machine Reading Comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 891–896.
- Wei He, Kai Liu, Jing Liu, Yajuan Lyu, Shiqi Zhao, Xinyan Xiao, Yuan Liu, Yizhong Wang, Hua Wu, Qiaoqiao She, Xuan Liu, Tian Wu, and Haifeng Wang. 2018. DuReader: a Chinese Machine Reading Comprehension Dataset from Real-world Applications. In *Proceedings of the Workshop on Machine Reading for Question Answering*. 37–46.
- Minghao Hu, Furu Wei, Yuxing Peng, Zhen Huang, Nan Yang, and Dongsheng Li. 2019. Read + Verify: Machine Reading Comprehension with Unanswerable Questions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 6529–6537.

- Rongtao Huang, BOWEI Zou, Yu Hong, Wei Zhang, Ai Ti Aw, and Guodong Zhou. 2020. NUT-RC: Noisy User-generated Text-oriented Reading Comprehension. In *Proceedings of the 28th International Conference on Computational Linguistics*. 2687–2698.
- Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. 2017. TriviaQA: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Vol. 1. 1601–1611.
- Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. In *ICLR 2020 : Eighth International Conference on Learning Representations*.
- Dongfang Li, Baotian Hu, Qingcai Chen, Weihua Peng, and Anqi Wang. 2020b. Towards Medical Machine Reading Comprehension with Structural Knowledge and Plain Text. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 1427–1438.
- Hongyu Li, Tengyang Chen, Shuting Bai, Takehito Utsuro, and Yasuhide Kawada. 2020a. MRC Examples Answerable by BERT without a Question Are Less Effective in MRC Model Training.. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing: Student Research Workshop*. 146–152.
- Xiaodong Liu, Yelong Shen, Kevin Duh, and Jianfeng Gao. 2018. Stochastic Answer Networks for Machine Reading Comprehension. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Vol. 1. 1694–1704.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv preprint arXiv:1907.11692* (2019).
- Siyu Long, Ran Wang, Kun Tao, Jiali Zeng, and Xinyu Dai. 2020. Synonym Knowledge Enhanced Reader for Chinese Idiom Reading Comprehension. In *Proceedings of the 28th International Conference on Computational Linguistics*. 3684–3695.
- Huashao Luo, Yu Shi, Ming Gong, Linjun Shou, and Tianrui Li. 2020. MaP: A Matrix-based Prediction Approach to Improve Span Extraction in Machine Reading Comprehension. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*. 687–695.
- Jonathan Malmaud, Roger Levy, and Yevgeni Berzak. 2020. Bridging Information-Seeking Human Gaze and Machine Reading Comprehension. In *Proceedings of the 24th Conference on Computational Natural Language Learning*. 142–152.
- Todor Mihaylov and Anette Frank. 2019. Discourse-Aware Semantic Self-Attention for Narrative Reading Comprehension. 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)*. 2541–2552.
- Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. MS MARCO: A Human Generated Machine Reading Comprehension Dataset.. In *CoCo@NIPS*.
- Kyosuke Nishida, Itsumi Saito, Kosuke Nishida, Kazutoshi Shinoda, Atsushi Otsuka, Hisako Asano, and Junji Tomita. 2019. Multi-style Generative Reading Comprehension. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 2273–2284.
- Wei Peng, Yue Hu, Luxi Xing, Yuqiang Xie, Jing Yu, Yajing Sun, and Xiangpeng Wei. 2020. Bi-directional Cognitive Thinking Network for Machine Reading Comprehension. In *Proceedings of the 28th International Conference on Computational Linguistics*.
- Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 1532–1543.
- Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, Vol. 1. 2227–2237.
- Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know What You Don’t Know: Unanswerable Questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, Vol. 2. 784–789.
- Min Joon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2016. Bidirectional Attention Flow for Machine Comprehension. In *ICLR (Poster)*.
- Kai Sun, Dian Yu, Dong Yu, and Claire Cardie. 2019. Improving Machine Reading Comprehension with General Reading Strategies. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. 2633–2643.
- Zhixing Tian, Yuanzhe Zhang, Kang Liu, Jun Zhao, Yantao Jia, and Zhicheng Sheng. 2020. Scene Restoring for Narrative Machine Reading Comprehension. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 3063–3073.

- Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer networks. In *NIPS'15 Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2*, Vol. 28. 2692–2700.
- Shuohang Wang and Jing Jiang. 2017. Machine Comprehension Using Match-LSTM and Answer Pointer. In *ICLR (Poster)*. 1.
- Wei Wang, Ming Yan, and Chen Wu. 2018c. Multi-Granularity Hierarchical Attention Fusion Networks for Reading Comprehension and Question Answering. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Vol. 1. 1705–1714.
- Wenhui Wang, Nan Yang, Furu Wei, Baobao Chang, and Ming Zhou. 2017. 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)*, Vol. 1. 189–198.
- Yizhong Wang, Kai Liu, Jing Liu, Wei He, Yajuan Lyu, Hua Wu, Sujian Li, and Haifeng Wang. 2018a. Multi-Passage Machine Reading Comprehension with Cross-Passage Answer Verification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Vol. 1. 1918–1927.
- Yucheng Wang, Bowen Yu, Yueyang Zhang, Tingwen Liu, Hongsong Zhu, and Limin Sun. 2020. TPLinker: Single-stage Joint Extraction of Entities and Relations Through Token Pair Linking. In *Proceedings of the 28th International Conference on Computational Linguistics*. Barcelona, Spain (Online), 1572–1582.
- Zhen Wang, Jiachen Liu, Xinyan Xiao, Yajuan Lyu, and Tian Wu. 2018b. Joint Training of Candidate Extraction and Answer Selection for Reading Comprehension. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Vol. 1. 1715–1724.
- Felix Wu, Ni Lao, John Blitzer, Guandao Yang, and Kilian Q. Weinberger. 2018. FAST READING COMPREHENSION WITH CONVNETS. *arXiv preprint arXiv:1711.04352* (2018).
- Yichong Xu, Xiaodong Liu, Yelong Shen, Jingjing Liu, and Jianfeng Gao. 2019. Multi-task Learning with Sample Re-weighting for Machine Reading Comprehension. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. 2644–2655.
- Ming Yan, Jiangnan Xia, Chen Wu, Bin Bi, Zhongzhou Zhao, Ji Zhang, Luo Si, Rui Wang, Wei Wang, and Haiqing Chen. 2019. A Deep Cascade Model for Multi-Document Reading Comprehension. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 7354–7361.
- An Yang, Quan Wang, Jing Liu, Kai Liu, Yajuan Lyu, Hua Wu, Qiaoqiao She, and Sujian Li. 2019b. Enhancing Pre-Trained Language Representations with Rich Knowledge for Machine Reading Comprehension. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 2346–2357.
- Wei Yang, Yuqing Xie, Aileen Lin, Xingyu Li, Luchen Tan, Kun Xiong, Ming Li, and Jimmy Lin. 2019c. End-to-End Open-Domain Question Answering with BERTserini. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*. 72–77.
- Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019a. XLNet: Generalized Autoregressive Pretraining for Language Understanding. In *Advances in Neural Information Processing Systems*, Vol. 32. 5753–5763.
- Adams Wei Yu, David Dohan, Minh-Thang Luong, Rui Zhao, Kai Chen, Mohammad Norouzi, and Quoc V. Le. 2018. QANet: Combining Local Convolution with Global Self-Attention for Reading Comprehension. In *International Conference on Learning Representations*.
- Yury Zemlyanskiy, Joshua Ainslie, Michiel de Jong, Philip Pham, Ilya Eckstein, and Fei Sha. 2021. ReadTwice: Reading Very Large Documents with Memories. *arXiv preprint arXiv:2105.04241* (2021).
- Xuemiao Zhang, Kun Zhou, Sirui Wang, Fuzheng Zhang, Zhongyuan Wang, and Junfei Liu. 2020c. Learn with Noisy Data via Unsupervised Loss Correction for Weakly Supervised Reading Comprehension. In *Proceedings of the 28th International Conference on Computational Linguistics*. 2624–2634.
- Zhuosheng Zhang, Yuwei Wu, Hai Zhao, Zuchao Li, Shuailiang Zhang, Xi Zhou, and Xiang Zhou. 2020a. Semantics-Aware BERT for Language Understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 9628–9635.
- Zhuosheng Zhang, Yuwei Wu, Junru Zhou, Sufeng Duan, Hai Zhao, and Rui Wang. 2020b. SG-Net: Syntax-Guided Machine Reading Comprehension. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 9636–9643.
- Zhuosheng Zhang, Junjie Yang, and Hai Zhao. 2021. Retrospective Reader for Machine Reading Comprehension. In *AAAI 2021*.
- Bo Zheng, Haoyang Wen, Yaobo Liang, Nan Duan, Wanxiang Che, Daxin Jiang, Ming Zhou, and Ting Liu. 2020. Document Modeling with Graph Attention Networks for Multi-grained Machine Reading Comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 6708–6718.