DCMN+: Dual Co-Matching Network for Multi-choice Reading Comprehension

Anonymous ACL submission

Abstract

Multi-choice reading comprehension is a challenging task to select an answer from a set of candidate options when given passage and question. Previous approaches usually only calculate question-aware passage representation and ignore passage-aware question representation when modeling the relationship between passage and question, which obviously cannot take the best of information between passage and question. In this work, we propose dual co-matching network (DCMN) which models the relationship among passage, question and answer options bidirectionally. Besides, inspired by how human solve multi-choice questions, we integrate two reading strategies into our model: (i) passage sentence selection that finds the most salient supporting sentences to answer the question, (ii) answer option interaction that encodes the comparison information between answer options. DCMN integrated with the two strategies (DCMN+) obtains state-of-the-art results on five multi-choice reading comprehension datasets which are from different domains: RACE, SemEval-2018 Task 11, ROCStories, COIN, MCTest.

1 Introduction

Machine reading comprehension (MRC) is a fundamental and long-standing goal of natural language understanding which aims to teach the machine to answer a question automatically according to a given passage (Hermann et al., 2015; Rajpurkar et al., 2016; Nguyen et al.). In this paper, we focus on multi-choice MRC task such as RACE (Lai et al., 2017) which requests to choose the right option from a set of candidate answers according to given passage and question. Different from MRC datasets such as SQuAD (Rajpurkar et al., 2016) and NewsQA (Trischler et al., 2017) where the expected answer is usually in the form of a short span from the given passage, answer in multi-choice MRC is non-extractive and may not appear in the original passage, which allows rich types of questions such as commonsense reasoning and passage summarization, as illustrated by the example question in Table 1.

Pre-trained language models such as BERT (Devlin et al.) and XLNet (Yang et al., 2019) have achieved significant improvement on various MRC tasks. Recent works on MRC may be put into two categories, training more powerful language models or exploring effective applying pattern of the language model to solve specific task. There is no doubt that training a better language model is essential and indeed extremely helpful (Devlin et al.; Yang et al., 2019) but at the same time it is time-consuming and resource-demanding to impart massive amounts of general knowledge from external corpora into a deep language model via pre-training (Sun et al.). For example, training a 24-layer transformer (Devlin et al.) requires 64 TPUs for four days (one year on eight P100 GPUs). So from the practical viewpoint, given limited computing resources and a well pre-trained model, can we improve the machine reading comprehension during fine-tuing instead of via expensive pre-training? This work starts from this viewpoint and focus on exploring effective applying pattern of language model instead of presenting better language models to further more enhance state-of-the-art multi-choice MRC. We will show the usage of a strong pre-trained language model may still have a heavy impact on MRC performance no matter how strong the language model itself is.

To well handle the multi-choice MRC problem, a common solution is to carefully model the relationship among the triplet of three sequences, passage (**P**), question (**Q**) and option (**A**) with a matching module to determine the answer. However, previous matching strategies fraught with ob-

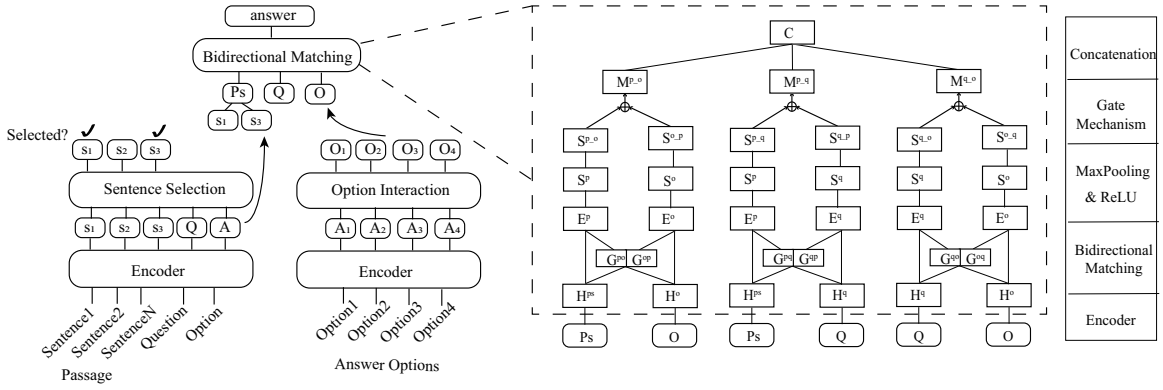

Figure 1: The framework of our model. P-Passage, Q-Question, O-Option.

Passage: Runners in a relay race pass a stick in one direction. However, merchants passed silk, gold, fruit, and glass along the Silk Road in more than one direction. They earned their living by traveling the famous Silk Road. ... **The Silk Road was made up of many routes, not one smooth path.** They passed through what are now 18 countries. The routes crossed mountains and deserts and had many dangers of hot sun, deep snow and even battles...
Question: The Silk Road became less important because _ . A. it was made up of different routes B. silk trading became less popular **C. sea travel provided easier routes** D. people needed fewer foreign goods

Table 1: An example passage with related question and options from RACE dataset. The ground-truth answer and the evidence sentences in the passage are in **bold**.

vious shortcomings (Wang et al., 2018; Tang et al., 2019; Chen et al., 2018; Ran et al., 2019) are usually unidirectional which only calculate question-aware passage representation and ignore passage-aware question representation when modeling the relationship between passage and question.

Thus, to alleviate such an obvious defect in modeling the {**P**, **Q**, **A**} triplet from existing work, we propose dual co-matching network (DCMN) which incorporates all the pairwise relationships among the {**P**, **Q**, **A**} triplet bidirectionally. In detail, we model the passage-question, passage-option and question-option pairwise relationship simultaneously and bidirectionally for each triplet and exploit the gated mechanism to fuse the representations from two directions. Besides, we integrate two reading strategies which humans usually use into the model. One is passage sentence selection that helps extract salient evidence sentences from the passage, and then matches evidence sentences with answer options. The other is answer option interaction that encodes comparison information into each option. The overall framework is shown in Figure 1. The output of language model (i.e. BERT (Devlin et al.) and XLNet (Yang et al., 2019)) is used as the contextual encoding. After passage sentence selection and answer option interaction, bidirectional matching representations are built for every pairwise relationship among the {**P**, **Q**, **A**} triplet.

Our model achieves new state-of-the-art results on the multi-choice MRC benchmark challenge RACE (Lai et al., 2017). We further conduct experiments on four representative multi-choice MRC datasets from different domains (i.e., ROCStories (Mostafazadeh et al., 2016), SemEval-2018 Task 11 (Ostermann et al., 2018), MCTest (Richardson et al., 2013), COIN Shared Task 1 (Ostermann et al., 2018)) and achieve the absolute improvement of 4.9% in average accuracy over directly fine-tuing BERT (2.8% on XLNet), which indicates our method has a heavy impact on the MRC performance no matter how strong the language model itself is.

2 Our Proposed Model

The illustration of our model is shown in Figure 1. The major components of the model are Contextual Encoding, Passage Sentence Selection, Answer Option Interaction and Bidirectional Matching. We will discuss each component in detail.

2.1 Task Definition

For the task of multi-choice reading comprehension, the machine is given a passage (**P**), a question (**Q**), and a set of answer options (**A**) and the goal is to select the correct answer from the candidates, where $\mathbf{P} = \{\mathbf{p_1}, \mathbf{p_2}, ..., \mathbf{p_n}\}$ is the passage composed of n sentences, $\mathbf{A} = \{\mathbf{A_1}, \mathbf{A_2}, ..., \mathbf{A_m}\}$ is the option set with m answer options.

2.2 Contextual Encoding

The large pre-trained language model has shown to be very powerful in language representation. In this work, the output of the pre-trained model is used as the contextual embedding which encodes each token in passage and question into a fixed-length vector. Given an encoder, the passage, the question, and the answer options are encoded as follows:

$$\mathbf{H}^p = Encode(\mathbf{P}), \mathbf{H}^q = Encode(\mathbf{Q}) \quad (1)$$

$$\mathbf{H}^a = Encode(\mathbf{A}) \quad (2)$$

where $Encode(\cdot)$ returns the last layer output in the encoder, which can be powerful strong pre-trained language models such as BERT (Devlin et al.) and XLNet (Yang et al., 2019). $\mathbf{H}^p \in R^{|P|\times l}$, $\mathbf{H}^q \in R^{|Q|\times l}$, and $\mathbf{H}^a \in R^{|A|\times l}$ are sequence representation of the passage, question and answer option, respectively. $|P|$, $|Q|$, $|A|$ are the sequence length of the passage, the question and the answer options, respectively. l is the dimension of the hidden state.

2.3 Passage Sentence Selection

Existing multi-choice MRC models focus on learning the passage representation with all the sentences which is inefficient and counter-intuitive. We study the number of sentences required to answer the question by randomly sampling 50 examples from the development set of RACE and COIN, as shown in Table 2. Among all examples, 87% questions on RACE and 86% on COIN can be answered within two sentences. From this observation, the model should be extremely beneficial if given one or two key evidence sentences.

To select the evidence sentences from the passage $\mathbf{P} = \{\mathbf{p_1}, \mathbf{p_2}, .., \mathbf{p_i}, .., \mathbf{p_n}\}$, this module scores each sentence p_i with respect to the question **Q** and answer option **A** in parallel. The top K scored sentences will be selected. This module shares the encoder with the whole model. For each $\{\mathbf{p_i}, \mathbf{Q}, \mathbf{A}\}$ triplet, we get $\mathbf{H}^{p_i} \in R^{|p_i|\times l}$, $\mathbf{H}^q$, and $\mathbf{H}^a$ from the contextual encoding layer. Here we introduce two methods to compute the score of the triplet based on the contextual encoding.

- **Cosine score**: The model computes word-by-word cosine distance between the sentence and question-option sequence pair. The score can be computed as follows:

$$\mathbf{D}^{pa} = Cosine(\mathbf{H}^a, \mathbf{H}^{p_i}) \in R^{|A|\times|p_i|} \quad (3)$$

$$\mathbf{D}^{pq} = Cosine(\mathbf{H}^q, \mathbf{H}^{p_i}) \in R^{|Q|\times|p_i|} \quad (4)$$

$$\bar{\mathbf{D}}^{pa} = MaxPooling(\mathbf{D}^{pa}) \in R^{|A|} \quad (5)$$

$$\bar{\mathbf{D}}^{pq} = MaxPooling(\mathbf{D}^{pq}) \in R^{|Q|} \quad (6)$$

$$score = \frac{\sum_{k=1}^{|A|} \bar{\mathbf{D}}_k^{pa}}{|A|} + \frac{\sum_{k=1}^{|Q|} \bar{\mathbf{D}}_k^{pq}}{|Q|} \quad (7)$$

where $\mathbf{D}^{pa}$, $\mathbf{D}^{pa}$ are the distance matrices and $\mathbf{D}_{ij}^{pa}$ is the cosine distance between the i-th word in the candidate option and the j-th word in the passage sentence.

- **Bilinear score**: Inspired by (Min et al., 2018a), we compute the bilinear weighted distance between two sequences, which can be calculated as follows:

$$\alpha = SoftMax(\mathbf{H}^q W_1) \in R^{|Q|\times l} \quad (8)$$

$$\mathbf{q} = \alpha^T \mathbf{H}^q \in R^l \quad (9)$$

$$\bar{\mathbf{P}}_j = \mathbf{H}_j^{p_i} W_2 \mathbf{q} \in R^l, j \in [1, |p_i|] \quad (10)$$

$$\hat{\mathbf{P}}^{pq} = Max(\bar{\mathbf{P}}_1, \bar{\mathbf{P}}_2, ..., \bar{\mathbf{P}}_{|p_i|}) \in R^l \quad (11)$$

where W_1, $W_2 \in R^{l\times l}$ are learnable parameters. $\hat{\mathbf{P}}^{pq}$ is the bilinear similarity vector between the passage sentence and question. This vector $\hat{\mathbf{P}}^{pa}$ between the passage sentence

N sent	% on RACE	% on COIN	Passage	Question
1	65	76	Soon after, the snack came out. *I then opened the chips and started to enjoy them, before enjoying the* ***soda***. I had a great little snack...	What else did the person enjoy?
2	22	10	*She lived in the house across the street that had 9 people and* ***3 dogs and another cat*** *living in it. She didn't seem very happy there, especially with a 2 year old that chased her and grabbed her.* The other people in the house agreed...	What did the 2 year old's mom own?
>=3	13	14	When I was hungry last week for a little snack and a soda, I went to the closest vending machine. I felt that it was a little overpriced, but being as though I needed something...	What's the main idea of this passage?

Table 2: Analysis of the sentences in passage required to answer questions on RACE and COIN. 50 examples from each dataset are sampled randomly. N sent indicates the number of sentences required to answer the question. The evidence sentences in the passage are in *emphasis* and the correct answer is with **bold**.

and answer can be calculated with the same procedure. The final score can be computed as follows:

$$score = W_3^T \hat{\mathbf{P}}^{pq} + W_4^T \hat{\mathbf{P}}^{pa} \quad (12)$$

where W_3, $W_4 \in R^l$ are learnable parameters.

After scoring each sentence, the top K scored sentences are selected[1] and concatenated together as the new passage $\mathbf{P}_s$ to replace original full passage. So the new sequence triplet is $\{\mathbf{P}_s, \mathbf{Q}, \mathbf{A}\}$ and the new passage encoding is represented as $\mathbf{H}^{ps}$.

2.4 Answer Option Interaction

Human solving multi-choice problem may seek help from comparing all answer options, for example, one option has to be picked up not because it is the most likely correct, but all the others are impossibly correct. Inspired by such human practice, we encode the comparison information into each answer option so that each option is not independent of the other. Here we build bilinear representations among options and encode the interaction information into each option. Gated mechanism (Srivastava et al.) is used to fuse the interaction representation with the original contextual encoding.

This module is also built on the contextual encoding layer which encodes each answer option $\mathbf{A}_i$ as $\mathbf{H}^{a_i}$. Then the comparison vector between option $\mathbf{A}_i$ and $\mathbf{A}_j$ can be computed as follows:

$$\mathbf{G} = SoftMax(\mathbf{H}^{a_i} W_5 \mathbf{H}^{a_j T}) \in R^{|A_i| \times |A_j|} \quad (13)$$

$$\mathbf{H}^{a_{i,j}} = ReLU(\mathbf{G}\mathbf{H}^{a_j}) \in R^{|A_i| \times l} \quad (14)$$

where $W_5 \in R^{l \times l}$ is one learnable parameter. $\mathbf{G}$ is the bilinear interaction matrix between A_i and A_j. $\mathbf{H}^{a_{i,j}}$ is the interaction representation. The interaction representations between A_i and other options $\{\mathbf{H}^{a_{i,j}}\}_{j \neq i}$ can be calculated by above equations. Then gated mechanism is used to fuse interaction representation with original contextual encoding as follows:

$$\hat{\mathbf{H}}^{a_i} = [\{\mathbf{H}^{a_{i,j}}\}_{j \neq i}] \in R^{|A_i| \times (m-1)l} \quad (15)$$

$$\bar{\mathbf{H}}^{a_i} = \hat{\mathbf{H}}^{a_i} W_6 \in R^{|A_i| \times l} \quad (16)$$

$$g = \sigma(\bar{\mathbf{H}}^{a_i} W_7 + \mathbf{H}^{a_i} W_8 + b) \quad (17)$$

$$\mathbf{H}^{o_i} = g * \mathbf{H}^{a_i} + (1 - g) * \bar{\mathbf{H}}^{a_i} \quad (18)$$

where W_7, $W_8 \in R^{l \times l}$ and $W_6 \in R^{(m-1)l \times l}$ are learnable parameters. $\hat{\mathbf{H}}^{a_i}$ is the concatenation of the interaction representations. $g \in R^{|A_i| \times l}$ is the reset gate which balances the influence of $\bar{\mathbf{H}}^{a_i}$ and $\mathbf{H}^{a_i}$. $\mathbf{H}^{o_i}$ is the final option representation of $\mathbf{A}_i$ encoded with the interaction information. So $\mathbf{O} = \{\mathbf{H}^{o_1}, \mathbf{H}^{o_2}, ..., \mathbf{H}^{o_m}\}$ is the final option representation set fused with contextual encoding and comparison information across answer options.

[1] K is the hyperparameter and will be discussed in the experiment part.

2.5 Bidirectional Matching

The triplet changes from {**P**, **Q**, **A**} to {$\mathbf{P}_s$, **Q**, **O**} after sentence selection and option interaction. To fully model the relationship in the {$\mathbf{P}_s$, **Q**, **O**} triplet, bidirectional matching is built to get all pairwise representations among the triplet, including passage-answer, passage-question and question-answer representation. Here shows how to modeling the relationship between question-answer sequence pair and it is the same for the other two pairs.

Bidirectional matching representation between the question $\mathbf{H}^q$ and answer option $\mathbf{H}^o$ can be calculated as follows:

$$\mathbf{G}^{qo} = SoftMax(\mathbf{H}^q W_9 \mathbf{H}^{oT}), \quad (19)$$

$$\mathbf{G}^{oq} = SoftMax(\mathbf{H}^o W_{10} \mathbf{H}^{qT}), \quad (20)$$

$$\mathbf{E}^q = \mathbf{G}^{qo}\mathbf{H}^o, \mathbf{E}^o = \mathbf{G}^{oq}\mathbf{H}^q, \quad (21)$$

$$\mathbf{S}^q = ReLU(\mathbf{E}^q W_{11}), \quad (22)$$

$$\mathbf{S}^o = ReLU(\mathbf{E}^o W_{12}), \quad (23)$$

where W_9, W_10, W_11, $W_12 \in R^{l\times l}$ are learnable parameters. $\mathbf{G}^{qo} \in R^{|Q|\times|O|}$ and $\mathbf{G}^{oq} \in R^{|O|\times|Q|}$ are the weight matrices between the question and candidate option. $\mathbf{E}^q \in R^{|Q|\times l}, \mathbf{E}^o \in R^{|A|\times l}$ represent option-aware question representation and question-aware option representation, respectively. The final representation of question-answer pair is calculated as follows:

$$\mathbf{S}^{q\text{-}o} = MaxPooling(\mathbf{S}^q), \quad (24)$$

$$\mathbf{S}^{o\text{-}q} = MaxPooling(\mathbf{S}^o), \quad (25)$$

$$g = \sigma(\mathbf{S}^{q\text{-}o} W_{13} + \mathbf{S}^{o\text{-}q} W_{14} + b), \quad (26)$$

$$\mathbf{M}^{q\text{-}o} = g * \mathbf{S}^{o\text{-}q} + (1-g) * \mathbf{S}^{o\text{-}q}, \quad (27)$$

where $W_{13}, W_{14} \in R^{l\times l}$ and $b \in R^l$ are three learnable parameters. After a row-wise max pooling operation, we get the aggregation representation $\mathbf{M}^q \in R^l$ and $\mathbf{M}^o \in R^l$. $g \in R^l$ is the reset gate. $\mathbf{M}^{q\text{-}o} \in R^l$ is the final bidirectional matching representation of the question-answer sequence pair.

Passage-question and passage-option sequence matching representation $\mathbf{M}^{p\text{-}q}, \mathbf{M}^{p\text{-}o} \in R^l$ can be calculated in the same procedure from Eq.(19) to Eq.(27). The framework of this module is shown in Figure 1.

2.6 Objective Function

In previous layer, we build the matching representation $\mathbf{M}^{p\text{-}q}, \mathbf{M}^{p\text{-}o}, \mathbf{M}^{q\text{-}o}$ for three sequence pairs. Finally, we concatenate them as the final representation $\mathbf{C} \in R^{3l}$ for each passage-question-option triplet. We can build $\mathbf{C}_i$ for each $\{P_s, Q, O_i\}$ triplet. If A_k is the correct option, then the objective function can be computed as follows:

$$\mathbf{C} = [\mathbf{M}^{p\text{-}q}; \mathbf{M}^{p\text{-}o}; \mathbf{M}^{q\text{-}o}], \quad (28)$$

$$L(A_k|P,Q) = -log\frac{exp(V^T\mathbf{C}_k)}{\sum_{j=1}^{m} exp(V^T\mathbf{C}_j)}, \quad (29)$$

where $V \in R^{3l}$ is a learnable parameter and m is the number of answer options.

3 Experiments

3.1 Dataset

We evaluate our model on five multi-choice MRC datasets from different domains. Statistics of these datasets are detailed in Table 3. Accuracy is calculated as $acc = N^+/N$, where N^+ and N are the number of correct predictions and the total number of questions. Some details about these datasets are shown as follows:

- **RACE** (Lai et al., 2017): RACE consists of two subsets: RACE-M and RACE-H corresponding to middle school and high school

Task	Domain	#o	#p	#q
RACE	general	4	27,933	**97,687**
SemEval	narrative text	2	2,119	13,939
ROCStories	stories	2	3472	3472
MCTest	stories	4	660	2,640
COIN	everyday scenarios	2	-	5,102

Table 3: Statistics of multi-choice machine reading comprehension datasets. #o is the average number of candidate options for each question. #p is the number of documents included in the dataset. #q indicates the total number of questions in the dataset.

difficulty level, which is recognized as one of the largest and most difficult datasets in multi-choice reading comprehension.

- **SemEval-2018 Task11** (Ostermann et al., 2018): In this task, systems are required to answer multi-choice questions based on narrative texts about everyday activities.

- **ROCStories** (Mostafazadeh et al., 2016): This dataset contains 98,162 five-sentence coherent stories in the training dataset (a large unlabeled stories dataset), 1,871 four-sentence story contexts along with a right ending and a wrong ending in the development and test datasets, respectively

- **MCTest** (Richardson et al., 2013): This task requires machines to answer questions about fictional stories, directly tackling the high-level goal of open-domain machine comprehension.

- **COIN Task 1** (Ostermann et al., 2018): The data for the task is short narrations about everyday scenarios with multiple-choice questions about them.

3.2 Implementation Details

We evaluate our proposed model based on the pre-trained language model BERT (Devlin et al.) and XLNet (Yang et al., 2019) which both have small and large versions. For example, the basic version BERT$_{base}$ has 12-layer transformer blocks, 768 hidden-size, and 12 self-attention heads, totally 110M parameters. The large version BERT$_{large}$ has 24-layer transformer blocks, 1024 hidden-size, and 16 self-attention heads, totally 340M parameters. XLNet also has a small and large version with similar size of BERT.

In our experiments, the max input sequence length is set to 512. A dropout rate of 0.1 is applied to every BERT layer. We optimize the model using BertAdam (Devlin et al.) optimizer with a learning rate 2e-5. We train for 10 epochs with batch size 8 using eight 1080Ti GPUs when BERT$_{large}$ and XLNet$_{large}$ are used as the encoder. Batch size is set to 16 when using BERT$_{base}$ and XLNet$_{base}$ as the encoder[2].

[2]Our implementation is based on https://github.com/huggingface/pytorch-transformers. Our code will be open after the blind review period ends.

Model	RACE-M/H	RACE
HAF (Zhu et al., 2018)	45.0/46.4	46.0
MRU (Tay et al., 2018)	57.7/47.4	50.4
HCM (Wang et al., 2018)	55.8/48.2	50.4
MMN (Tang et al., 2019)	61.1/52.2	54.7
GPT (Radford, 2018)	62.9/57.4	59.0
RSM (Sun et al.)	69.2/61.5	63.8
OCN (Ran et al., 2019)	76.7/69.6	71.7
XLNet (Yang et al., 2019)	85.5/80.2	81.8
BERT$_{base}$*	71.1/62.3	65.0
BERT$_{large}$*	76.6/70.1	72.0
XLNet$_{large}$*	83.7/78.6	80.1
Our Models		
BERT$_{base}$* + DCMN	73.2/64.2	67.0
BERT$_{large}$* + DCMN	79.2/72.1	74.1
BERT$_{large}$* + DCMN + P$_{SS}$ + A$_{OI}$	79.3/74.4	**75.8**
XLNet$_{large}$* + DCMN + P$_{SS}$ + A$_{OI}$	86.5/81.3	**82.8**
Human Performance		
Turkers	85.1/69.4	73.3
Ceiling	95.4/94.2	94.5

Table 4: Experiment results on RACE test set. All the results are from single models. P$_{SS}$: Passage Sentence Selection; A$_{OI}$: Answer Option Interaction. * indicates our implementation.

3.3 Evaluation on RACE

In Table 4, we report the experimental results on RACE and its two subtasks: RACE-M and RACE-H. In the table, Turkers is the performance of Amazon Turkers on a randomly sampled subset of the RACE test set and Ceiling is the percentage of the unambiguous questions with a correct answer in a subset of the test set. Here we give the results of directly fine-tuning BERT$_{base}$, BERT$_{large}$ and XLNet$_{large}$ on RACE and get the accuracy of 65.0%, 72.0% and 80.1%, respectively. Because of the limited computing resources, the largest batch size can only be set to 8 in our experiments which leads to 1.7% decrease (80.1% vs. 81.8%) on XLNet compared to the result reported in (Yang et al., 2019)[3].

From the comparison, we observe that our proposed method obtains significant improvement over directly fine-tuning language models (75.8% vs. 72.0% on BERT$_{large}$ and 82.8% vs. 80.1% on XLNet$_{large}$) achieves the state-of-the-art result on RCAE.

3.4 Ablation Study on RACE

In Table 5, we focus on the contribution of main components (DCMN, passage sentence selection and answer option interaction) in our model. From the results, we see that the bidirectional match-

[3]The implementation is very close to the result 80.3% in (Yang et al., 2019) when using batch size 8 on RACE.

	BERT$_{base}$	BERT$_{large}$	XLNet$_{large}$
base encoder	64.6	71.8	80.1
+ DCMN	66.0 (+1.4)	73.8 (+2.0)	81.5 (+1.4)
+ DCMN + P$_{SS}$	66.6 (+2.0)	74.6 (+2.8)	82.1 (+2.0)
+ DCMN + A$_{OI}$	66.8 (+2.2)	74.4 (+2.6)	82.2 (+2.1)
+ ALL(DCMN+)	**67.4 (+2.8)**	**75.4 (+3.6)**	**82.6 (+2.5)**

Table 5: Ablation study on RACE dev set. P$_{SS}$: Passage Sentence Selection. A$_{OI}$: Answer Option Interaction. DCMN+: DCMN + P$_{SS}$ + A$_{OI}$

ing strategy (DCMN) gives the main contribution and achieves further improvement by integrating with reading strategies. Finally, we obtain the best performance by combining all components (DCMN+).

3.5 Evaluation on Other Multi-choice Datasets

We further conduct experiments on four other multi-choice MRC datasets and the results are shown in Table 6. When adapting the method to the non-conventional MRC dataset ROCStories which requires to choose the correct ending to a four-sentence incomplete story from two answer options (Mostafazadeh et al., 2016), the question context is left empty as no explicit questions are provided. Passage sentence selection is not used in this dataset because there are only four sentences as the passage. Since the test set of COIN is not publicly available, we report the performance of the model on the development set. For other tasks, we select the model that achieves the highest accuracy on development set and report the accuracy on test set.

As shown in Table 6, we achieve state-of-the-art (SOTA) results on five datasets and obtain 3.1% absolute improvement in average accuracy over the previous average SOTA (88.9% vs. 85.8%) by using BERT as encoder, 4.8% (90.6% vs. 85.8%) by using XLNet as encoder. To further investigate the contribution of our model, we also report the results that directly fine-tune BERT/XLNet on the target datasets. From the comparison, we can see that our model obtains 4.9%, 2.8% absolute improvement in average accuracy over directly fine-tuing BERT (88.9% vs. 84.0%) and XLNet (90.6% vs. 87.8%), respectively. These results indicate our proposed model has a heavy impact on the performance no matter how strong the language model itself is.

[4]Here we omit the combinations with $S^{O\text{-}P}$ because we find the combinations with $S^{P\text{-}O}$ works better than $S^{O\text{-}P}$.

3.6 Unidirectional vs. Bidirectional

Here we mainly focus on whether the bidirectional matching strategy works better than previous unidirectional method. In Table 7, we enumerate all the combinations of unidirectional matching strategies and show the matching methods of HCM (Wang et al., 2018) and HAF (Zhu et al., 2018) using our annotation. From the comparison, we observe that all bidirectional combinations works better than the unidirectional ones which include the matching methods in HCM and HAF. All the pairwise matching representations ($M^{P\text{-}Q}$, $M^{P\text{-}O}$, $M^{Q\text{-}O}$) are necessary and by concatenating them together, we achieve the highest performance (67.1%).

3.7 Evaluation on RACE and COIN with Different Settings in Passage Sentence Selection

Table 8 shows the performance comparison with different scoring methods, and we observe that both methods have their advantages and disadvantages. Cosine score method works better on COIN dataset (83.5% vs. 82.8%) and bilinear score works better on RACE dataset (66.8% vs. 66.5%).

Figure 2 shows the results of passage sentence selection on COIN and RACE dev set with different number of selected sentences (Top K). The results without the sentence selection module are also shown in the figure (RACE-w and COIN-w) for comparison. We observe that sentence selection consistently shows a positive impact on both datasets when more than four sentences are selected compared to the model without sentence selection (RACE-w and COIN-w). The highest performance is achieved when top 3 sentences are selected on COIN and top 5 sentences on RACE where the main reason is that the questions in RACE are designed by human experts and require more complex reasoning.

3.8 Does Our Model Works Better than Previous Methods?

As shown in Table 4, applying previous models to BERT (i.e., BERT+HCM and BERT+MMN) show no performance increase than directly fine-tuning BERT. The contrast is clear that our proposed model achieves more than 3.8% absolute increase compared to BERT baseline. We summarize the reasons resulting in such contrast as

Task	Previous STOA		BERT	BERT+DCMN+	XLNet	XLNet+DCMN+
SemEval Task 11	(Sun et al.)	89.5	90.5	91.8 (+1.3)	92.0	93.4 (+1.4)
ROCStories	(Li et al., 2019)	91.8	90.8	92.4 (+1.6)	93.8	95.8 (+2.0)
MCTest-MC160	(Sun et al.)	81.7	73.8	85.0 (+11.2)	80.6	86.2 (+5.6)
MCTest-MC500	(Sun et al.)	82.0	80.4	86.5 (+6.1)	83.4	86.6 (+3.2)
COIN Task 1	(Devlin et al.)	84.2	84.3	88.8 (+4.5)	89.1	91.1 (+2.0)
Average		85.8	84.0	**88.9 (+4.9)**	87.8	**90.6 (+2.8)**

Table 6: Results on the test set of SemEval Task 11, ROCStories, MCTest and the development set of COIN Task 1. The test set of COIN is not public. DCMN+: DCMN + P_{SS} + A_{OI}. Previous SOTA: previous state-of-the-art model. All the results are from single models.

Model	RACE	Model	RACE	Model	RACE
base encoder	64.6				
+ Unidirectional					
$[S^{P\text{-}O}; S^{P\text{-}Q}; S^{O\text{-}Q}]$	65.0	$[S^{P\text{-}Q}; S^{Q\text{-}O}]$	63.4	$[S^{P\text{-}Q}; S^{O\text{-}Q}]$	64.5
$[S^{P\text{-}O}; S^{Q\text{-}P}; S^{O\text{-}Q}]$	65.2	$[S^{P\text{-}O}; S^{Q\text{-}O}]$	63.6	$[S^{Q\text{-}P}; S^{O\text{-}Q}]$	65.2
$[S^{P\text{-}Q}; S^{P\text{-}O}]$ (Wang et al., 2018) (HCM)	65.4	$[S^{P\text{-}O}; S^{O\text{-}Q}]$	64.2	$[S^{P\text{-}Q}; S^{O\text{-}P}]$	64.7
$[S^{P\text{-}O}; S^{P\text{-}Q}; S^{Q\text{-}O}]$ (Zhu et al., 2018) (HAF)	64.2	$[S^{P\text{-}O}; S^{Q\text{-}P}; S^{Q\text{-}O}]$	64.4	$[S^{Q\text{-}P}; S^{Q\text{-}O}]$	64.3
$[S^{Q\text{-}O}; S^{O\text{-}Q}; S^{P\text{-}Q}; S^{P\text{-}O}]$ (Tang et al., 2019) (MMN)	63.2				
+ Bidirectional					
$[M^{P\text{-}Q}; M^{P\text{-}O}]$	66.4	$[M^{P\text{-}O}; M^{Q\text{-}O}]$	66.0	$[M^{P\text{-}Q}; M^{Q\text{-}O}]$	65.5
$[M^{P\text{-}Q}; M^{P\text{-}O}; M^{Q\text{-}O}]$ (DCMN)	**67.1**				

Table 7: Performance comparison with different combination methods on the RACE dev set[4]. We use BERT_{base} as our encoder here. $[;]$ indicates the concatenation operation. $S^{P\text{-}O}$ is the unidirectional matching referred in Eq. 24. $M^{P\text{-}O}$ is the bidirectional matching representation referred in Eq. 27. Here uses our annotations to show previous matching strategies.

Top K	1	2	3	4	5	6
RACE-cos	58.4	60.1	63.3	65.8	**66.5**	66
RACE-bi	89.5	60.5	63.4	**66.8**	66.4	66.2
COIN-cos	81.0	82.0	**83.5**	83.0	82.5	82.4
COIN-bi	81.7	82.0	82.6	**82.8**	82.4	82.2

Table 8: Results on RACE and COIN dev set with different scoring methods (cosine and bilinear score in P_{SS}). We use BERT_{base} as encoder here.

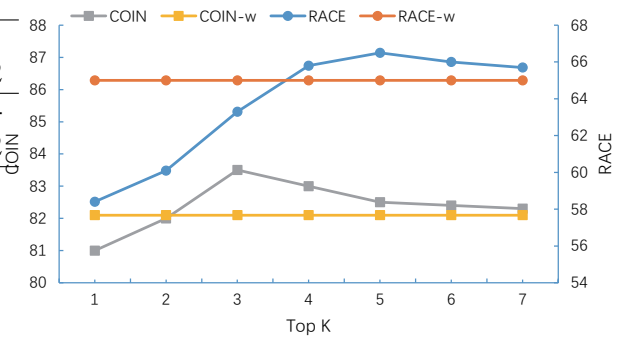

Figure 2: Results of sentence selection on the development set of RACE and COIN when selecting different number of sentences (Top K). We use BERT_{base} as encoder and cosine score method here. RACE/COIN-w indicates the results on RACE/COIN without sentence selection module.

follows: (i) the unidirectional representations can not take the best use of the information between two sequences, (ii) previous methods (Wang et al., 2018; Tang et al., 2019) use elementwise subtraction and multiplication to fuse $\mathbf{E}^q$ and $\mathbf{H}^o$ in Eq. 21 (i.e., $[\mathbf{E}^q \ominus \mathbf{H}^o; \mathbf{E}^q \otimes \mathbf{H}^o]$) which is shown not good enough as such processing breaks the symmetry of equation. Symmetric representations from both directions show essentially helpful for our bidirectional architecture.

3.9 Evaluation on Different Types of Questions

Inspired by (Sun et al.), we further analyze the performance of the main components on different question types. Questions are roughly divided into five categories: detail (*facts and details*), inference

(*reasoning ability*), main (*main idea*), attitude (*authors attitude toward a topic*) and vocabulary (*vocabulary questions*) (Lai et al., 2017; Qian and Schedl, 2004). We annotate all the instances of the RACE development set. As shown in Figure 3, all the combinations of components works better than directly fine-tuning BERT in most question types. Bidirectional matching strategy (DCMN) consistently improves the results across all categories. DCMN+P_{SS} works best on the inference and attitude categories which indicates the sentence selection module improves the reasoning ability of the model. DCMN+A_{OI} works better than DCMN on detail and main categories which indicates that the model achieves better distinguish ability with answer option interaction module.

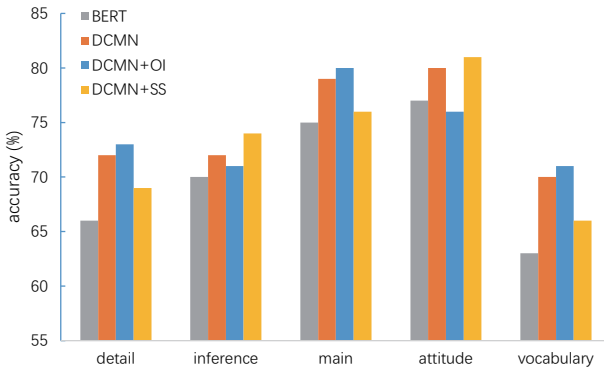

Figure 3: Performance on different question types, tested on the RACE development set. BERT$_{large}$ is used as encoder here. OI: Answer Option Interaction. SS: Passage Sentence Selection.

4 Related Work

The task of selecting sentences to answer the question has been studied across several QA datasets (Min et al., 2018b; Wang et al., 2019; Choi et al., 2017). (Wang et al., 2019) apply distant supervision to generate imperfect labels and then use them to train a neural evidence extractor. (Min et al., 2018b) propose a simple sentence selector to select the minimal set of sentences then feed into the QA model. They are different from our work in that (i) we select the sentences by modeling the relevance among sentence-question-option triplet, not sentence-question pair. (ii) Our model uses the output of language model as the sentence embedding and computes the relevance score using these sentence vectors directly, without the need of manually defined labels. (iii) We achieve a generally positive impact by selecting sentences while previous sentence selection methods usually bring performance decrease in most cases.

Most recent works attempting to integrate comparison information focus on building attention mechanism at word-level (Ran et al., 2019; Zhu et al., 2018) where the performance increase is very limited. Our answer option interaction module is different from previous works in that: (i) we encode the comparison information by modeling the bilinear representation among the options at sentence-level which is similar to the procedure of modeling passage-question sequence relationship, without attention mechanism. (ii) We use gated mechanism to fuse the comparison information with the original contextual encoding.

5 Conclusion

This paper proposes dual co-matching network integrated with two reading strategies (passage sentence selection and answer option interaction) to improve the reading comprehension ability of machine. By combining our model with BERT and XLNet[5], we obtain performance increase by a large margin over directly fine-tuning BERT/XLNet and achieve state-of-the-art results on five representative multi-choice MRC datasets including RACE. The experiment results consistently indicate the general effectiveness and applicability of our model.

References

Zhipeng Chen, Yiming Cui, Wentao Ma, and Shijin Wang. 2018. Convolutional Spatial Attention Model for Reading Comprehension with Multiple-Choice Questions. volume abs/1811.08610.

Eunsol Choi, Daniel Hewlett, Jakob Uszkoreit, Illia Polosukhin, Alexandre Lacoste, and Jonathan Berant. 2017. Coarse-to-fine question answering for long documents. In *Proceedings of ACL 2017*, pages 209–220, Vancouver, Canada.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *CoRR*, abs/1810.04805.

Karl Moritz Hermann, Tomás Kociský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching Machines to Read and Comprehend. *CoRR*, abs/1506.03340.

[5]Note that our model can be easily adapted to other pre-trained language models such as RoBERTa (**?**) and SpanBERT (Joshi et al., 2019) (not released).

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2019. Spanbert: Improving pre-training by representing and predicting spans. *CoRR*, abs/1907.10529.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. 2017. RACE: Large-scale ReAding Comprehension Dataset From Examinations. In *Proceedings of EMNLP 2017*, pages 785–794.

Zhongyang Li, Xiao Ding, and Ting Liu. 2019. Story ending prediction by transferable BERT. *CoRR*, abs/1905.07504.

Sewon Min, Victor Zhong, Richard Socher, and Caiming Xiong. 2018a. Efficient and robust question answering from minimal context over documents. In *Proceedings of the ACL 2018*, pages 1725–1735.

Sewon Min, Victor Zhong, Richard Socher, and Caiming Xiong. 2018b. Efficient and Robust Question Answering from Minimal Context over Documents. In *Proceedings of ACL 2018*, pages 1725–1735.

Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen. 2016. A corpus and cloze evaluation for deeper understanding of commonsense stories. In *Proceedings of NAACL 2016*, pages 839–849.

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. MS MARCO: A Human Generated MAchine Reading COmprehension Dataset. *CoRR*, abs/1611.09268.

Simon Ostermann, Michael Roth, Ashutosh Modi, Stefan Thater, and Manfred Pinkal. 2018. SemEval-2018 task 11: Machine comprehension using commonsense knowledge. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 747–757, New Orleans, Louisiana.

David Qian and Mary Schedl. 2004. Evaluation of an in-depth vocabulary knowledge measure for assessing reading performance. *Language Testing - LANG TEST*, 21:28–52.

Alec Radford. 2018. Improving Language Understanding by Generative Pre-Training.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ Questions for Machine Comprehension of Text. In *Proceedings of EMNLP 2016*, pages 2383–2392.

Qiu Ran, Peng Li, Weiwei Hu, and Jie Zhou. 2019. Option comparison network for multiple-choice reading comprehension. *CoRR*, abs/1903.03033.

Matthew Richardson, Christopher J.C. Burges, and Erin Renshaw. 2013. MCTest: A challenge dataset for the open-domain machine comprehension of text. In *Proceedings of EMNLP 2013*, pages 193–203.

Rupesh Kumar Srivastava, Klaus Greff, and Jürgen Schmidhuber. Highway networks. *CoRR*, abs/1505.00387.

Kai Sun, Dian Yu, Dong Yu, and Claire Cardie. Improving Machine Reading Comprehension with General Reading Strategies. *CoRR*, abs/1810.13441.

Min Tang, Jiaran Cai, Hankz Hankui Zhuo, and Hankz Hankui Zhuo. 2019. Multi-Matching Network for Multiple Choice Reading Comprehension. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Yi Tay, Luu Anh Tuan, and Siu Cheung Hui. 2018. Multi-range Reasoning for Machine Comprehension. *CoRR*, abs/1803.09074.

Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman. 2017. NewsQA: A Machine Comprehension Dataset. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 191–200. Association for Computational Linguistics.

Hai Wang, Dian Yu, Kai Sun, Jianshu Chen, Dong Yu, Dan Roth, and David A. McAllester. 2019. Evidence sentence extraction for machine reading comprehension. *CoRR*, abs/1902.08852.

Shuohang Wang, Mo Yu, Jing Jiang, and Shiyu Chang. 2018. A Co-Matching Model for Multi-choice Reading Comprehension. In *Proceedings of ACL 2018*, pages 746–751.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *CoRR*, abs/1906.08237.

Haichao Zhu, Furu Wei, Bing Qin, and Ting Liu. 2018. Hierarchical Attention Flow for Multiple-choice Reading Comprehension. In *Thirty-Second AAAI Conference on Artificial Intelligence*.