Context-guided Triple Matching for Multiple Choice Question Answering

Anonymous ACL submission

Abstract

The task of multiple choice question answering (MCQA) is to identify the correct answer from multiple candidates given a passage and a question. It is typically approached by estimating the matching score among the *triple* of the passage, question and candidate answers. Existing methods decouple this estimation into several pairwise or *dual* matching steps, that limited the ability of assessing cases with the subtle difference from candidate answers. This paper introduces a **C**ontext-guided **T**riple **M**atching algorithm, which models the matching among the triple simultaneously. Precisely, the proposed matching takes one component from the triple as the context, and estimates its semantic matching between the other two. Additionally, a contrastive term is adopted to model the dissimilarity between the correct answer and distractive ones. The proposed algorithm is validated on several benchmarking MCQA datasets and outperforms the state-of-the-art models by a large margin.

1 Introduction

Question answering is one of the most popular and challenging research topics in machine reading comprehension (MRC). Existing studies of question answering focus on either *extracting* spans (a short but continuous sequence of words) from the given passage (Seonwoo et al., 2020; Joshi et al., 2020) or *selecting* the correct answer from a set of candidate answers, known as multiple choice question answering (MCQA) (Duan et al., 2021; Li et al., 2021; Zhang et al., 2020a).

Approaches to MCQA usually consist of a two-step process. In the first step, words in the **triple** (*i.e.* passage (p), question (q) and answer(a)) are encoded (usually by pre-trained language models) into fixed length of vectors. Typical models include Bert(Devlin et al., 2019), Roberta(Liu et al., 2019), and XLNet(Yang et al., 2019), *etc*. The second step is to utilize those vector representation and further match semantically among the triple (Chaturvedi et al., 2018; Wang et al., 2018; Zhang et al., 2020b; Zhu et al., 2022). In the recent work of *DCMN+*(Zhang et al., 2020a), the conventional unidirectional matching (Chaturvedi et al., 2018; Wang et al., 2018) is extended to a bidirectional one among the pairs of (p, q), (p, a), and (q, a), respectively. The bidirectional matching improves the capability of capturing the semantic relationship among the triple, so as the performance, compared with the previous unidirectional matching. Yet, such bidirectional methods only consider the interaction between two components from the triple, which has limited its ability to handle cases with the presence of subtle differences from candidate answers.

Table 1: An example from the RACE dataset. The evidence sentences (from the passage) and keywords (from the question and candidate answers) are highlighted. DCMN+ picks answer A where the correct answer is B.

Question: According to the passage, when we become **adults**, _____?
Passage: Most people believe they don't have imagination. ··· but most of us, **once we became adults**, **forget how to access it**. Creativity isn't always connected with great works of art or ideas. **People at work and in their free time routinely think of creative ways to solve problems**. ··· **Here are three techniques to help you.** ···
Answers: *A*. most of us are no longer **creative**; *B*. we can still learn to be more **creative**; *C*. we are not as imaginative as children; *D*. we are unwilling to be **creative**;

Table 1 shows an illustrative example from the popular MCQA dataset (RACE (Lai et al., 2017)). As observed, candidate answers contain lexically same keyword (*i.e.* "creative"). Yet, with the existing pairwise/dual strategy, only two components are matched at one time. As such, a matching of (p, a) or (q, a) is hardly distinguishable given similar candidates a (lacking of the third component q or p); additionally, the matching of (p, q)

is the same across all candidate answers (without the third component a). Therefore, answering this cloze question requires providing a context (*i.e.* the third component) in performing the conventional pairwise matching.

This paper accordingly proposes a novel **C**ontext-guided **T**riple **M**atching (CTM). First, we extend the conventional pairwise to a triple matching by employing one component from the triple as the prior context. In other words, the proposed matching is performed to match between two components semantically with respect to this prior context, so the entire triple is matched at one time. Second, we further adopt a contrastive regularization in capturing the subtle semantic differences among answer candidates. The purpose is to maximize the similarity of features from correct triple(s) while pushing away that of distractive ones, that has been neglected by existing methods.

We summarize the contributions of this paper as follows[1]:

- context is introduced into the matching process, and a context-guided triplet matching is proposed accordingly to improve the ability in effectively capturing semantic relationship from a passage, questions and answers; and
- contrastive regularization is utilized to enhance subtle semantic differences among similar candidate answers; and
- extensive experiments are conducted on two widely used MCQA datasets to evaluate the proposed CTM, and state-of-the-art results are achieved (an improvement of approximately 2.5 percentage points) in comparison with existing methods.

2 Related work

Multiple choice question answering (MCQA) is a long-standing research problem from machine reading comprehension, where the key is to determine one correct answer (from all candidates) given the background passage and question. Several models have been proposed to utilize deep neural networks with different **matching** strategies.

Chaturvedi *et al.* first concatenate the question and candidate answer, and calculate the matching degree against the passage via attention (Chaturvedi et al., 2018). The work (Wang et al., 2018) treats the question and a candidate answer as two sequences before matching them individually with the given passage. Then a hierarchical aggregation structure is constructed to fuse the previous co-matching representation to predict answers. Similarly, a hierarchical attention flow is proposed in (Zhu et al., 2018) to estimate the matching relationship based on the attention mechanism at different hierarchical levels. Zhang *et al.* propose a dual co-matching network in (Zhang et al., 2020a), which formulates the matching model among background passages, questions, and answers bi-bidirectionally.

Apart from the aforementioned matching-based work, another line of studies proposes to integrate with the auxiliary knowledge. For instance, a syntax-enhanced network is presented in (Zhang et al., 2020b) to combine syntactic tree information with the pre-trained encoder for better linguistic matching. Duan *et al.* utilize the semantic role labeling to enhance the contextual representation before modeling the correlation (Duan et al., 2021). More recently, the off-the-shelf knowledge graph is leveraged to fine-tune the downstream MCQA task in (Li et al., 2021).

Compared to existing matching work in (Wang et al., 2018; Zhang et al., 2020a), the proposed algorithm performs matching by introducing a context (an entity from the triple of passage, question and answer). This context serves as a background knowledge to exploit the semantic relationship with the remaining two entities.

3 Proposed method

The proposed method gradually identifies the best-matching answer by coordinating the loss from Triple Matching (*TM*) and Contrastive Regularization (*CR*) simultaneously, as illustrated in Fig. 1.

Given an input triple of passage, question and answer, a pre-trained language model is first utilized for *encoding* textual contents. Then the TM module enumerates this input triple and selects one entity as the background context. The semantic relationship is accordingly estimated using the remaining two entities with regard to this selected context. At last, the produced features from TM are utilized for answer *selection*, while CR ensures feature *enhancement* so that the feature similarity between correct triples is maximized, by contrasting to that from distractive ones.

[1] The source codes will be made publicly available from Github.

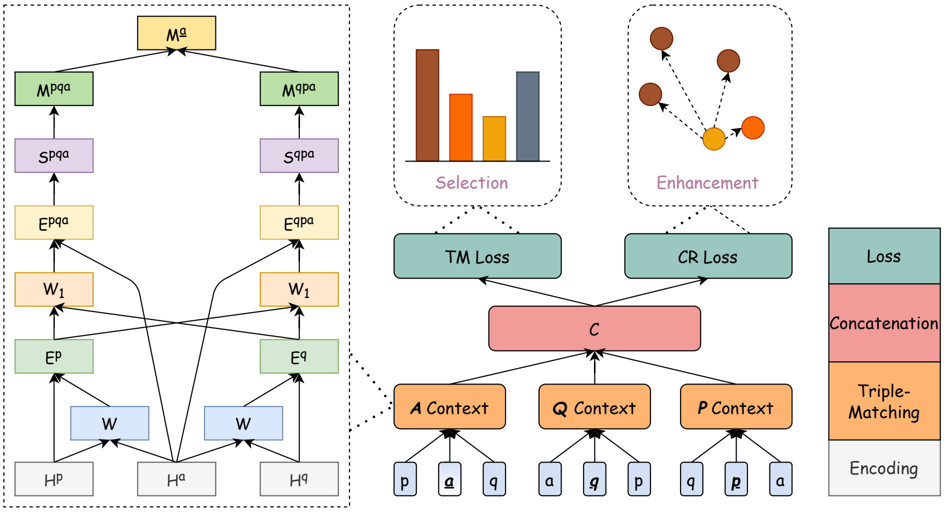

Figure 1: Overview of the proposed **C**ontext-guided **T**riple **M**atching algorithm for MCQA, which is characterized by a Triple Matching (TM) and Contrastive Regularization (CR) module. Taking answer-guided passage-question matching ($\boldsymbol{M}^{\underline{a}}$) as an example, TM aims to estimate the semantic correlation between the selected $\underline{a}$ with anther two entities (p and q). CR further captures the subtle difference (or highlights the feature dissimilarity) between correct and distractive answers.

3.1 Encoding

Let p, q and a be a passage, a question and a candidate answer, respectively. A pre-trained model (*e.g.* $BERT$) is adopted to encode each word from them into a fixed-length vector, yielding

$$\boldsymbol{H}^p = Enc(p), \boldsymbol{H}^q = Enc(q), \boldsymbol{H}^a = Enc(a), \tag{1}$$

where $\boldsymbol{H}^p \in \mathbb{R}^{|p|\times l}$,$\boldsymbol{H}^q \in \mathbb{R}^{|q|\times l}$,and $\boldsymbol{H}^a \in \mathbb{R}^{|a|\times l}$ are relevant representation of p, q, and a, respectively, and l is the dimension of the hidden state.

3.2 Triple matching

To model the relationship among the triple of {p, q, a}, in TM we introduce an context-oriented mechanism. That is, we select one component from the triple once (as the background context), and estimate its semantic correlation with the remaining two to further produce a context-guided representation. Note that this proposed module involve all three entities from the triple simultaneously, while existing methods adopt the pairwise strategy that involves only two entities once.

Taking the answer a as an example, below we show how to model the representation for the answer(context)-guided passage-question matching. At first, given the encoder output of $\boldsymbol{H}^p$, $\boldsymbol{H}^a$ and $\boldsymbol{H}^q$, we apply the bidirectional attention to calculate the answer-aware passage representation ($\boldsymbol{E}^p \in \mathbb{R}^{|a|\times l}$) and answer-aware question representation ($\boldsymbol{E}^q \in \mathbb{R}^{|a|\times l}$) as follows:

$$\begin{aligned} \boldsymbol{G}^{aq} &= SoftMax(\boldsymbol{H}^a W \boldsymbol{H}^{qT}), \boldsymbol{E}^p = \boldsymbol{G}^{ap}\boldsymbol{H}^p \\ \boldsymbol{G}^{ap} &= SoftMax(\boldsymbol{H}^a W \boldsymbol{H}^{pT}), \boldsymbol{E}^q = \boldsymbol{G}^{aq}\boldsymbol{H}^q, \end{aligned} \tag{2}$$

where $W \in \mathbb{R}^{l\times l}$ are learnable parameters, and $\boldsymbol{G}^{aq} \in \mathbb{R}^{|a|\times|q|}$ and $\boldsymbol{G}^{ap} \in \mathbb{R}^{|a|\times|p|}$ are the attention matrix between the answer-question, and the answer-passage, respectively.

Next, we further allow the third entity to be included by adopting the bidirectional attention again (to embed the question for $\boldsymbol{E}^p$ and the passage for $\boldsymbol{E}^q$). As a result, the core of triple matching becomes:

$$\begin{aligned} \boldsymbol{G}^{pq} &= SoftMax(\boldsymbol{E}^p W_1 \boldsymbol{E}^{qT}) \\ \boldsymbol{G}^{qp} &= SoftMax(\boldsymbol{E}^q W_1 \boldsymbol{E}^{pT}) \\ \boldsymbol{E}^{pqa} &= \boldsymbol{G}^{pq}\boldsymbol{H}^a, \boldsymbol{E}^{qpa} = \boldsymbol{G}^{qp}\boldsymbol{H}^a \\ \boldsymbol{S}^{pqa} &= ReLU(\boldsymbol{E}^{pqa}W_2), \boldsymbol{S}^{qpa} = ReLU(\boldsymbol{E}^{qpa}W_2), \end{aligned} \tag{3}$$

where W_1, $W_2 \in \mathbb{R}^{l\times l}$ are learnable parameters, and $\boldsymbol{E}^{pqa} \in \mathbb{R}^{|a|\times l}$, $\boldsymbol{E}^{qpa} \in \mathbb{R}^{|a|\times l}$ represent passage-question-aware answer representation and question-passage-aware answer representation, respectively. The final representation of answer-guided passage-question matching (*i.e.* $\boldsymbol{M}^{\underline{a}} \in$

$\mathbb{R}^{2\times l}$) is to aggregate the above as follows:

$$\begin{aligned}\boldsymbol{M}^{pqa} &= MaxPooling(\boldsymbol{S}^{pqa})\\ \boldsymbol{M}^{qpa} &= MaxPooling(\boldsymbol{S}^{qpa})\\ \boldsymbol{M}^{\underline{a}} &= [\boldsymbol{M}^{pqa}; \boldsymbol{M}^{qpa}].\end{aligned} \tag{4}$$

In sum, the proposed TM module for answer-guided passage-question matching $\boldsymbol{M}^{\underline{a}}$ is illustrated on the left of Figure 1. Similarly, we enumerate the other two entities (that is, the question q and passage p) to compute the related representation for the question-guided answer-passage matching (*ie.*, $\boldsymbol{M}^{\underline{q}} \in \mathbb{R}^{2\times l}$) and the passage-guided answer-question matching (*ie.*, $\boldsymbol{M}^{\underline{p}} \in \mathbb{R}^{2\times l}$), following the same procedure from Eq.(2) to Eq.(4).

3.3 Answer selection

With the triple-matching representations $\boldsymbol{M}^{\underline{a}}$, $\boldsymbol{M}^{\underline{q}}$, $\boldsymbol{M}^{\underline{p}}$, we further concatenate them as the final representation $\boldsymbol{C}$ (*ie.*, $\boldsymbol{C} = [\boldsymbol{M}^{\underline{a}}; \boldsymbol{M}^{\underline{q}}; \boldsymbol{M}^{\underline{p}}]$). Let $\boldsymbol{C}_c$ be the representation for the correct triple of $\{p, q, a_c\}$. Accordingly, the selection loss can be computed as follows:

$$\mathcal{L}_{TM}(p, q, a_c) = -log\frac{\exp(\boldsymbol{C}_c^T V)}{\sum_{\boldsymbol{C}_i \in \boldsymbol{C}_{\mathcal{S}}} \exp(\boldsymbol{C}_i^T V)}, \tag{5}$$

where $V \in \mathbb{R}^l$ is a learnable parameter, $\mathcal{S}$ is the set of all candidate answers, and $\boldsymbol{C}_{\mathcal{S}}$ is the feature set for $\mathcal{S}$.

3.4 Contrastive regularization as enhancement

The aforementioned TM module is performed to extract semantic representation from one candidate triple. Yet, there could be trivial (word) difference between the correct and distractive answers (as shown from Table 1). To highlight this dissimilarity, we accordingly utilize a contrastive regularization as a feature enhancement strategy.

More precisely, for the given passage p, the question q, the set of candidate answers $\mathcal{S}$, and the correct answer a_c, we aim to construct a group of positive (correct) triples (such as $\{p, q, a_c\}$) and another group of negative (wrong) triples ($\{p, q, a_w\}$), where $a_w \in \mathcal{S}$ and $a_w \neq a_c$. Notably, MCQA is enjoyed owing to those distractive answers, which in nature form negative triples against the correct ones. Then the proposed contrastive regularization is to encourage the latent representation from correct triples staying closer to each other while pushing away those distractive ones.

Furthermore, let $\boldsymbol{C}_c$ and $\boldsymbol{C}_w$ be the encoded representation of $\{p, q, a_c\}$ and $\{p, q, a_w\}$ using the TM module. To form the feature of another positive triple, we adopt the dropout-based operation (say $Drop(\cdot)$) from (Gao et al., 2021; Liang et al., 2021), which has been proven to be an effective way of creating similar feature. That is, we simply apply the TM module *twice* with different *dropout* masks to produce the representation of another positive triple, say $\boldsymbol{C}_c^+ = Drop(\boldsymbol{C}_c)$. Similarly, one could produce the negative feature via $\boldsymbol{C}_c^- = Drop(\boldsymbol{C}_w)$. Accordingly, the CR is defined using the negative log likelihood (NLL) loss as follows:

$$\mathcal{L}_{CR}(p, q, a_c) = -log\frac{\exp(\boldsymbol{C}_c^T \boldsymbol{C}_c^+/\tau)}{\sum_{\boldsymbol{C}_i \in \boldsymbol{C}_{\mathcal{S}/c} \cup \boldsymbol{C}_c^+ \cup \boldsymbol{C}_c^-} \exp(\boldsymbol{C}_c^T \boldsymbol{C}_i/\tau)}, \tag{6}$$

where τ is a pre-defined temperature. Notably, with the presence of $\boldsymbol{C}_c^-$, $\boldsymbol{C}_{\mathcal{S}}$ from Eq. (5) will be reformulated as $\boldsymbol{C}_{\mathcal{S}} = \boldsymbol{C}_{\mathcal{S}} \cup \boldsymbol{C}_c^-$, which is equivalent to increasing the number of wrong answers.

3.5 Loss function

With two losses from the answer selection and contrastive regularization, we propose to train the model using the *joint* loss as follows:

$$\mathcal{L} = \mathcal{L}_{TM} + \lambda_{CR}\mathcal{L}_{CR}, \tag{7}$$

where λ_{CR} is a penalty term[2].

3.6 Discussion

Next, we analyze the relationship between the proposed method and existing pairwise algorithms. Previous studies measure the matching representation (*i.e.* $\boldsymbol{C}$ from Eq. (5)) using the following estimation:

- CNN-Matching (Chaturvedi et al., 2018):

$$\begin{aligned}\boldsymbol{H}^{qa} &= Enc([q; a]); \boldsymbol{H}^p = Enc(p);\\ \boldsymbol{M} &= Att(\boldsymbol{H}^{qa}, \boldsymbol{H}^p);\\ \boldsymbol{C} &= Sim(\boldsymbol{H}^{qa}, \boldsymbol{M}).\end{aligned} \tag{8}$$

[2] There are another two training strategies, including *pre-train* and *alternate*. The former is to update the model first using $\mathcal{L}_{CR}$ before finetuning with $\mathcal{L}_{TM}$, while the latter is to train the model with $\mathcal{L}_{TM}$ for $(N_t - 1)$ iterations and switch to $\mathcal{L}_{CR}$ once, for every N_t iterations. However, the experimental results show the *joint* training outperforms *pre-train* and *alternate* based model.

- Co-Matching (Wang et al., 2018):

$$\begin{aligned}\boldsymbol{H}^q =& Enc(q); \boldsymbol{H}^a = Enc(a); \boldsymbol{H}^p = Enc(p);\\ \boldsymbol{M}^{qp} =& Att(\boldsymbol{H}^q, \boldsymbol{H}^p); \boldsymbol{M}^{ap} = Att(\boldsymbol{H}^a, \boldsymbol{H}^p);\\ \boldsymbol{C} =& [Sim(\boldsymbol{M}^{qp}, \boldsymbol{H}^p); Sim(\boldsymbol{M}^{ap}, \boldsymbol{H}^p)].\end{aligned} \tag{9}$$

- DCMN+ (Zhang et al., 2020a):

$$\begin{aligned}\boldsymbol{H}^q =& Enc(q); \boldsymbol{H}^a = Enc(a); \boldsymbol{H}^p = Enc(p);\\ \boldsymbol{M}^{qa} =& Att(\boldsymbol{H}^q, \boldsymbol{H}^a); \boldsymbol{M}^{qp} = Att(\boldsymbol{H}^q, \boldsymbol{H}^p);\\ \boldsymbol{M}^{ap} =& Att(\boldsymbol{H}^a, \boldsymbol{H}^p);\\ \boldsymbol{C} =& [Gat(\boldsymbol{M}^{qa}, \boldsymbol{M}^{ap}); Gat(\boldsymbol{M}^{qp}, \boldsymbol{M}^{ap});\\ & Gat(\boldsymbol{M}^{qa}, \boldsymbol{M}^{qp})].\end{aligned} \tag{10}$$

Within aforementioned methods, Enc represents the encoder, Att stands for the attention operation, Sim is for the similarity calculation, Gat is a rest gate function, and $[;]$ is the vector concatenation. Note that existing methods adopted different implementation of Enc, Att, and Sim, *etc*. For instance, Enc in (Chaturvedi et al., 2018) and (Wang et al., 2018) has been implemented as CNN and BERT, respectively.

Compared to the aforementioned methods, the proposed algorithm can be cast as their extension, with an additional consideration of triple matching and contrastively representing the correct answer(s). That is, the triple matching is to apply two attention layers to estimate the semantic relationship with regard to the selected context. As such, Eq.(2) to Eq.(4) can be equivalently represented as the following process:

$$\begin{aligned}\boldsymbol{M}^{qa} =& Att(\boldsymbol{H}^q, \boldsymbol{H}^a); \boldsymbol{M}^{pa} = Att(\boldsymbol{H}^p, \boldsymbol{H}^a);\\ \boldsymbol{M}^{pqa} =& Att(Att(\boldsymbol{M}^{pa}, \boldsymbol{M}^{qa}), \boldsymbol{H}^a);\\ \boldsymbol{M}^{qpa} =& Att(Att(\boldsymbol{M}^{qa}, \boldsymbol{M}^{pa}), \boldsymbol{H}^a).\end{aligned} \tag{11}$$

In addition, our method is also distinct from existing ones by further integrating the contrastive loss. That is, we aim to distinguish the correct answers via pulling its relevant representation away from distractive ones, which has been neglected by existing pairwise-matching approaches.

4 Experiments

4.1 Datasets

Two datasets adopted in the experiments are RACE (Lai et al., 2017) and DREAM (Sun et al., 2019). RACE is one of the widely used banchemark datasets for MCQA, which consists of subsets RACE-M and RACE-H that correspond to the reading-difficulty level of middle and high school, respectively. DREAM is a dialogue-based examination dataset. It includes dialog passages as the background and three options associated with each individual question. Their statistics are shown in Table 2.

Table 2: Summary of RACE and DREAM, where **#a** is the averaged number of candidate answers per question, and **#w/a** is the averaged length per answer.

Dataset	passages	Questions	#a	#w/a
RACE-M	7,139	28,293	4	4.9
RACE-H	20,794	69,394	4	6.8
DREAM	6,444	10,197	3	5.3

4.2 Implementation and settings

Two pre-trained language models, including the BERT$_{base}$ and BERT$_{large}$, are adopted as the encoder for word-embedding. BERT$_{base}$ consists of 12-layer transformer blocks, 12 self-attention heads, and 768 hidden-size, whereas BERT$_{large}$ consists of 24-layer transformer blocks, 16 self-attention heads, and 1024 hidden-size. They have 110M and 340M parameters, respectively. The dropout rate for each BERT layer is set as 0.1. The Adam optimizer with a learning rate setting of $2e^{-5}$ is adopted to train the proposed CTM.

During training, batch size is 4, number of training epoch is 3, and the max length of input sequences is set to 360 for RACE. For DREAM, batch size is 4 and number of training epochs is 6, and the max length of input sequences is set to 300. For passages with more words, the sliding-window strategy in (Jin et al., 2020) is adopted to split them into appropriate-length chunks. For the contrastive regularization, the dropout rate is 0.1 to produce one positive and one additional negative, and the temperature $\tau = 0.07$. The CTM model is trained on a machine with four Tesla K80 GPUs. Accuracy $acc = n_q^+/n_q$ is used to measure the performance, that is the ratio between correct-answered questions (n_q^+) and total questions (n_q).

4.3 Results

We compared the performance of the proposed CTM with the methods, including the public models from the leaderboard (*i.e.* BERT) and state-of-the-arts (*i.e.* DCMN+(Zhang et al., 2020a)). To make a fair comparison, we are particularly interested in those implemented with the same BERT encoder[3].

Results of the proposed CTM and comparing methods are shown in Table 3. The proposed method achieves state-of-the-art performance on both RACE and DREAM datasets. Not surprisingly, the BERT$_{base}$ methods achieve generally worse performance compared to the counterparts using BERT$_{large}$, which shows the benefit from a better pre-trained model. Although baseline performance (from BERTs) is further improved by bidirectional matching (Zhang et al., 2020a) or external knowledge (Zhang et al., 2020b), these strategies applied pairwise matching among the passage, question and answer independently, without considering the third entity from the triple. By contrast, the proposed CTM method utilize one entity as the background context to match with the other two, so that learned features are geared towards this selected context. The use of contrastive regularization further strengthens the learning to differentiate the correct answer from semantically closed but wrong ones. As a result, the proposed CTM substantially outperforms existing methods.

Table 3: Results in accuracy (%),obtained by CTM and the comparing methods on the on the test set. In the table, "-B" and "-L" represents the *base* and *large* model of the BERT encoder. "×" indicates there is no results from the original reference, and "★" shows the original reference doesn't differentiate RACE-M and RACE-H but only report the averaged result.

Algorithm	RACE-M	RACE-H	DREAM
BERT(-B)	71.1	62.3	63.2
BERT(-L)	76.6	70.1	66.8
DCMN+(-B)	73.2	64.2	×
DCMN+(-L)	79.3	74.4	×
SG-Net(-L)	78.8	72.2	×
CSFN(-B)★	68.3	68.3	64.0
DUMA(-B)	×	×	64.0
ConceptPlug(-B)★	65.3	65.3	65.3
ConceptPlug(-L)★	72.6	72.6	69.3
CTM(-B)	75.2	68.3	69.2
CTM(-L)	81.5	75.3	72.0

4.4 Ablation study

Experiments are conducted on the RACE dataset (with the BERT-base encoder) to validate the contribution from the proposed TM and CR loss.

On triple-matching This experiment compares the performance of two different matching strategies, *i.e.* the proposed TM against existing dual one. The contrastive regularization in this experiment is disabled by setting $\lambda_{CR} = 0$.

The DCMN+ model (Zhang et al., 2020a) is adopted as the opponent, which achieves the state-of-the-art performance. It consists of three dual-matching components: question-answer pair ($\boldsymbol{M}^{qa}$), question-passage pair ($\boldsymbol{M}^{qp}$), and answer-passage pair ($\boldsymbol{M}^{ap}$). By contrast, the proposed CTM includes three components, including $\boldsymbol{M}^{\underline{a}}$, $\boldsymbol{M}^{\underline{p}}$, $\boldsymbol{M}^{\underline{q}}$, respectively (see Eq. (4)). Next, we carefully ablate those components by enumerating different combinations, and compare them with DCMN+ using the RACE-H dataset.

Table 4: The performance comparison of the proposed CTM against DCMN+ on the RACE-H testing set, by employing combination of different pairwise matching.

Branch	Acc	Branch	Acc
$[\boldsymbol{M}^{ap}; \boldsymbol{M}^{qp}]$	63.8	$[\boldsymbol{M}^{ap}; \boldsymbol{M}^{qa}]$	63.3
$[\boldsymbol{M}^{qa}; \boldsymbol{M}^{qp}]$	62.1	$[\boldsymbol{M}^{ap}; \boldsymbol{M}^{qa}; \boldsymbol{M}^{qp}]$	64.2
$\boldsymbol{M}^{\underline{a}}$	48.6	$\boldsymbol{M}^{\underline{q}}$	64.1
$\boldsymbol{M}^{\underline{p}}$	63.8	$[\boldsymbol{M}^{\underline{a}}, \boldsymbol{M}^{\underline{q}}]$	64.6
$[\boldsymbol{M}^{\underline{p}}, \boldsymbol{M}^{\underline{q}}]$	64.8	$[\boldsymbol{M}^{\underline{a}}, \boldsymbol{M}^{\underline{p}}]$	52.3
$[\boldsymbol{M}^{\underline{a}}, \boldsymbol{M}^{\underline{p}}, \boldsymbol{M}^{\underline{q}}]$	65.7		

Table 4 shows the results on the proposed TM and DCMN+ (Zhang et al., 2020a). As observed, the component of $\boldsymbol{M}^{\underline{q}}$ contributes mostly in the answer selection, as it achieves the highest accuracy among all proposed components. This result suggests the importance of utilizing question(s) as the background context, rather than passage and/or answers, to address MCQA tasks.

On the other hand, the component of $\boldsymbol{M}^{\underline{a}}$ obtains the worst performance, which reveals the limitation of *short* answers. Note that the $\boldsymbol{M}^{\underline{a}}$ is to take the answer as the background context, and estimate its correlation (using attention) between passage/question. Yet, the attention-based correlation is insignificant compared to others, mainly due to the short sequence length from answers.

Additionally, we also notice that the combination of all three component work best with the encoder, which demonstrates a better matching outcome

[3] Given the availability of numerous pre-training models, we could simply replace the adopted BERTs with other more powerful encoders, such as (Liu et al., 2019), to improve the performance of CTM. Alternatively, to use the additional ground knowledge, such as the work (Zhang et al., 2020b), could also lead to a potential improvement for CTM. We leave these as the future work.

(65.7%) compared to that of DCMN+ (64.2%). The result not only indicates the necessity of utilizing all three proposed matching components, but also shows the superiority of the triple matching compared to existing dual matching.

On contrastive regularization The impact of the contrastive regularization is mainly controlled by the penalty term λ_{CR} from Eq. (7). We evaluate the accuracy by setting different values to λ_{CR} as [0, 0.5, 1, 1.5]. Notably, with $\lambda_{CR} = 0$, the model degrades to the simple TM module.

With the comparison result presented in Fig 2, we found out that the proposed CR helps in enhancing the matching capability (*i.e.*, via maximizing the feature difference between the correct and wrong triple). For instance, the CTM achieves the best result when $\lambda_{CR} = 0.5$, compared to that of $\lambda_{CR} = 0$. Yet, the increase of the λ_{CR} value results in the inferior accuracy (in particular with $\lambda_{CR} = 1.5$). The reason could be the compatibility between the learned features and the final classification. With a larger λ_{CR}, the model tends to learn distinct features to separate answers, which might not be useful for selecting the correct answer.

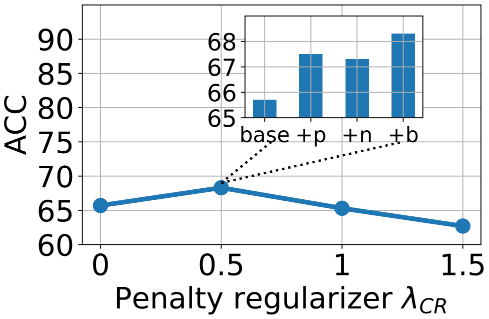

Figure 2: Performance comparison at different values of λ_{CR}.

Additionally, fixing $\lambda_{CR} = 0.5$, we further perform the ablation study on the created positive/negative to manifest its efficacy and the results are illustrated in the zoom-in area from Fig 2. In particular, "base" represents the simple TM module, "+p" considers to only add one positive $\boldsymbol{C}_c^+$ without additional negatives (*i.e.*, $\boldsymbol{C}_c^- = \emptyset$ in Eq. (6)), while "+n" represents the case of creating only one extra negative ($\boldsymbol{C}_c^-$, and $\boldsymbol{C}_c^+ = \emptyset$) so $\boldsymbol{C}_S$ from Eq. (5) is reformulated as $\boldsymbol{C}_S = \boldsymbol{C}_S \cup \boldsymbol{C}_c^-$. The model of "+b" applies both the positive and one extra negative sample.

The results clearly show contributions from individual aspect to the final performance. For instance, adding extra negative sample (associated with the case of "+n") enforces the model to learn a better representation for the correct triple, as the increasing number of wrong answers. On the other hand, adding $\boldsymbol{C}_c^+$ is in favor of the model via awarding the answer difference (or pushing away wrong triples from the correct ones). The combination of adding $\boldsymbol{C}_c^+$ and $\boldsymbol{C}_c^-$ outperforms other model variants, including the simple TM module, that evidently states the effectiveness of the proposed CR loss.

Analysis In this section, the model capability is further analyzed based on the question complexity. Followed by (Zhang et al., 2020a), we randomly select 10% samples (350 questions) from the testing set of RACE-H, and manually annotate them using question types of *what*, *which*, *cloze* and *other*[4]. Additionally, we further tag them based on the number of sentence required to answer the question. The performance from two models is accordingly shown in Table 5.

Table 5: Comparison of both the model performance on the RACE-H testing set, where cases are categorized by question types and the number of required sentences (**#s**). The underlined results (%) are from CTM, while the one within the bracket (%) represents that of DCMN+.

Type	**#s=1**	**#s=2**	**#s**$\geqslant 3$
what	69.2(68.3)	65.7(63.2)	62.5(58.3)
which	67.0(65.8)	67.9(64.6)	66.5(62.2)
cloze	70.2(63.3)	70.5(58.1)	66.0(55.8)
other	71.3(71.5)	68.7(64.6)	60.5(60.3)

The result clearly indicates the superiority of the proposed algorithm when answering complex questions, such as cloze test and more sentences involved. For instance, the cloze test requires more reasoning capability as the model needs to scan the entire passage according to the given question and all candidate answers. As such, the proposed triple matching is more suitable than the conventional dual-wise strategy. Additionally, as the cloze test needs to fill in missing item(s), the textual difference from candidate answers also plays an important role. As expected, the proposed contrastive regularization helps in capturing those difference, thereby achieving the improvement for the question answering.

[4] The "other" type including the rest question types, such as *why*, *who*, *when*, *where*, and *how*.

Similarly, with complex questions that need to infer from (more than) 3 sentences, the result clearly reflects an improvement from CTM compared with DCMN+. With the increasing number of required sentences, the prediction accuracy from both models has been reduced. Yet, CTM performs much stable than its counterpart, which shows its robustness of handling cases with multiple evidence sentences. In conclusion, it can be empirically confirmed that the proposed CTM algorithm achieves comparative performance than dual-wise methods, in particular with complex question answering.

5 Conclusion

The task of multiple choice question answering (MCQA) aims to identify a suitable answer from the background passage and question. Using the dual-based matching strategy, existing methods decouple the process into several pairwise steps, that fail to capture the global correlation from the triple of passage, question and answer.

In this paper, the proposed algorithm introduces a context-guided triple matching. Concretely, a triple-matching module is used to enumerate the triple and estimate a semantic matching between one component (context) with the other two. Additionally, to produce more informative features, the contrasitve regularization is further introduced to encourage the latent representation of correct triples staying away from distractive ones. Intensive experiments based on two benchmarking datasets are considered. In comparison to multiple existing approaches, the proposed algorithm produces a state-of-the-art performance by achieving higher accuracy. To our knowledge, this is the first work that explores a context-guided matching in multiple choice question answering. We will continue exploring inter/cross sentence matching as our future work.

References

Akshay Chaturvedi, Onkar Pandit, and Utpal Garain. 2018. CNN for text-based multiple choice question answering. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 272–277.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*, pages 4171–4186.

Qianwei Duan, Jun Huang, and Huiyan Wu. 2021. Contextual and semantic fusion network for multiple-choice reading comprehension. *IEEE Access*, 9:51669–51678.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple contrastive learning of sentence embeddings. *arXiv preprint arXiv:2104.08821*.

Di Jin, Shuyang Gao, Jiun-Yu Kao, Tagyoung Chung, and Dilek Hakkani-tur. 2020. MMM: Multi-stage multi-task learning for multi-choice reading comprehension. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8010–8017.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2020. SpanBERT: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. 2017. RACE: Large-scale ReAding comprehension dataset from examinations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 785–794.

Ronghan Li, Zejun Jiang, Lifang Wang, Xinyu Lu, Meng Zhao, and Daqing Chen. 2021. Enhancing Transformer-based language models with commonsense representations for knowledge-driven machine comprehension. *Knowledge-Based Systems*, 220:106936.

Xiaobo Liang, Lijun Wu, Juntao Li, Yue Wang, Qi Meng, Tao Qin, Wei Chen, Min Zhang, and Tie-Yan Liu. 2021. R-Drop: Regularized dropout for neural networks. *arXiv preprint arXiv:2106.14448*.

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. *CoRR*, abs/1907.11692.

Yeon Seonwoo, Ji-Hoon Kim, Jung-Woo Ha, and Alice Oh. 2020. Context-aware answer extraction in question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2418–2428.

Kai Sun, Dian Yu, Jianshu Chen, Dong Yu, Yejin Choi, and Claire Cardie. 2019. DREAM: A challenge dataset and models for dialogue-based reading comprehension. *Transactions of the Association for Computational Linguistics*, 7:217–231.

Shuohang Wang, Mo Yu, Jing Jiang, and Shiyu Chang. 2018. A co-matching model for multi-choice reading comprehension. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pages 746–751.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. XLNet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems*, volume 32.

Shuailiang Zhang, Hai Zhao, Yuwei Wu, Zhuosheng Zhang, Xi Zhou, and Xiang Zhou. 2020a. DCMN+: Dual co-matching network for multi-choice reading comprehension. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):9563–9570.

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 Thirty-Fourth AAAI Conference on Artificial Intelligence*, volume 34, pages 9636–9643.

Haichao Zhu, Furu Wei, Bing Qin, and Ting Liu. 2018. Hierarchical attention flow for multiple-choice reading comprehension. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1):6077–6084.

Pengfei Zhu, Zhuosheng Zhang, Hai Zhao, and Xiaoguang Li. 2022. DUMA: Reading comprehension with transposition thinking. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 30:269–279.