Context-guided Triple Matching for Multiple Choice Question Answering

Xun Yao, Junlong Ma, Xinrong Hu, and Junping Liu
School of Computer Science and Artificial Intelligence
Wuhan Textile University
Wuhan, China
{yaoxun, 2015363059, hxr, jpliu}@wtu.edu.cn

Hailun Wang and Gang Huang
College of Electrical and Information Engineering
Quzhou University
Quzhou, China
{wanghl, huangg}@qzc.edu.cn

Jie Yang and Wanqing Li
School of Computing and Information Technology
University of Wollongong
Wollongong, Australia
{jiey, wanqing}@uow.edu.au

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 pairwise or dual matching ignoring the third component. This paper introduces a Context-guided Triple Matching algorithm, which models the matching among the triple simultaneously. Specifically, 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.

Index Terms—Multiple Choice Question Answering, Triple matching, Contrastive regularization, Reasoning

I. 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 [1], [2], identifying entities from an external knowledge base [3], [4] or selecting the correct answer from a set of candidate answers, known as multiple choice question answering (MCQA) [5]–[7]. This paper is on a novel method for MCQA.

Approaches to MCQA usually consist of a two-step process. In the first step, tokens from the triple (i.e., a collection of components with passage (p), question (q) and answer(a)) are encoded (usually by pre-trained language models) into fixed length of vectors. Typical models include BERT [8], RoBerta [9], and Albert [10], etc. The second step is to utilize those vector representation and further match semantically among the triple [11]–[14]. A significant research effort, therefore, has been devoted to achieve high MCQA performance via fine-tuning pre-trained model(s) and/or improving the subsequent matching.

Recently, the conventional semantic matching has been extended from the unidirectional manner [11], [12] to a bidirectional way (such as DCMN+ [7]) among the pairs of (p, q), (p, a), and (q, a), respectively. This bidirectional or pairwise matching improves the capability of capturing the semantic relationship among the triple, so as the performance, compared with the previous unidirectional matching.

Table I

| Question (or q): |
|------------------|
| According to the passage, when we become adults, ____? |
| Passage (or p): |
| 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 (or a): |
| 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; |

Yet, pairwise methods only consider the interaction between two components from the triple. Such a matching of (p, q) is the same across all candidate answers (without taking into account the third component a). Similarly, the matching of (p, a) (or (q, a)) is hardly distinguishable with the presence of subtle differences across candidate answers a. Table I shows an illustrative example from the popular MCQA dataset (RACE [15]). As observed, candidate answers contain lexically same
keyword (i.e. “creative”). Due to the similar pairwise-matching scores across four candidate answers, the bidirectional method (DCMN+) picks the wrong answer. Therefore, we argue that answering MCQA questions requires the third component in performing the conventional pairwise matching.

This paper accordingly proposes a novel Context-guided Triple Matching (CTM), while the third component missing from the pairwise matching is adopted as a prior context. The proposed triple matching is present as a hierarchical attention flow to adequately capture the semantic relationship. Specifically, given a candidate triple, we first employ (any) one component from the triple as the prior context. Then we apply the bidirectional attention to calculate the correlation between context and the other two components separately. Afterwards, another attention layer is utilized to leverage two above correlations to form an aggregated context-aware representation. In this way, the model is able to gather more comprehensive semantic relationship for the triple, according to the selected context. Similarly, we enumerate the other two components (from the triple) and cast as the prior context to repeat the same attention flow. Finally, a fully-connected layer is employed for all formed context-aware representations to estimate the matching score. In addition to the triple matching, we also consider to adopt a contrastive regularization in capturing the subtle semantic differences among answer candidates. The aim 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.

The contributions of this paper is summarized as follows:

- context is introduced into the matching process, and a context-guided triplet matching is proposed accordingly via a hierarchical attention flow. The context is leveraged as an additional information, which has been ignored by previous pairwise works, to effectively capture semantic relationship from a passage, questions and answers; and contrastive regularization is further utilized to differentiate semantic discrepancy 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.

II. 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 [11]. The work [12] 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 [16] to estimate the matching relationship based on the attention mechanism at different hierarchical levels. Zhang et al. propose a dual co-matching network in [7], which formulates the matching model among background passages, questions, and answers bi-directionally.

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 [13] 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 [5]. More recently, the off-the-shelf knowledge graph is leveraged to fine-tune the downstream MCQA task in [6].

Compared to existing matching work in [7], [12], the proposed algorithm performs matching by introducing a context (an component from the triple of passage, question and answer). This context serves as a background knowledge to exploit the semantic relationship between the remaining two components.

III. 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 component as the background context. The semantic relationship is accordingly estimated using the remaining two components 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.

A. 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 in them into a fixed-length vector, yielding

\[
H^p = \text{Enc}(p), \quad H^q = \text{Enc}(q), \quad H^a = \text{Enc}(a),
\]

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

B. Triple matching

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

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 \(H^p, H^q\) and \(H^a\), we apply the bidirectional attention to calculate the answer-aware passage representation (\(E^p \in \mathbb{R}^{[a] \times l}\)) and question-passage-aware representation (\(E^q \in \mathbb{R}^{[a] \times l}\)) as follows:

\[
G^{aq} = \text{SoftMax}(H^a W H^q T), E^p = G^{ap} H^p \\
G^{ap} = \text{SoftMax}(H^p W H^q T), E^q = G^{aq} H^q,
\]

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

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

\[
G^{pa} = \text{SoftMax}(E^p W_1 E^q T) \\
G^{aq} = \text{SoftMax}(E^q W_1 E^p T) \\
E^{ppa} = G^{pa} H^a \\
E^{qpa} = G^{aq} H^p \\
S^{ppa} = \text{ReLU}(E^{ppa} W_2) \\
S^{qpa} = \text{ReLU}(E^{qpa} W_2),
\]

where \(W_1, W_2 \in \mathbb{R}^{l \times l}\) are learnable parameters, and \(E^{ppa} \in \mathbb{R}^{[a] \times l}\) represent passage-question-aware answer representation and question-passage-aware answer representation, respectively. The final representation of answer-guided passage-question matching (i.e., \(M^2 \in \mathbb{R}^{2 \times l}\)) is to aggregate the above the as follows:

\[
M^{ppa} = \text{MaxPooling}(S^{ppa}) \\
M^{qpa} = \text{MaxPooling}(S^{qpa}) \\
M^2 = [M^{ppa}; M^{qpa}].
\]

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

C. Answer selection

With the triple-matching representations \(M^2, M^2, M^2\), we further concatenate them as the final representation \(C\) (i.e., \(C = [M^2; M^2; M^2]\)). Let \(C_c\) be the representation for the correct triple of \(\{p, q, a_c\}\). Accordingly, the selection loss can be computed as follows:

\[
\mathcal{L}_{TM}(p, q, a_c) = -\log \frac{\exp(C_c^T V)}{\sum_{C_i \in C_S} \exp(C_i^T V)},
\]

where \(V \in \mathbb{R}^{l}\) is a learnable parameter, \(S\) is the set of all candidate answers, and \(C_S\) is the feature set for \(S\).

D. 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
dissimilarity, we accordingly utilize a contrastive regularization as a feature enhancement strategy.

Specifically, 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 $C_c$ and $C_w$ 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 approach (say $\text{Drop}(\cdot)$) from [17], [18], which has 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 $C_c^+ = \text{Drop}(C_c)$. Similarly, one could produce the negative feature via $C_c^- = \text{Drop}(C_w)$. Accordingly, the CR is defined using the negative log likelihood (NLL) loss as follows:

$$L_{CR}(p, q, a_c) = -\log \exp(C_c^T C_c^+ / \tau) \sum_{C_i \in C_S \cup C_c^+ \cup C_c^-} \exp(C_i^T C_i / \tau),$$

(6)

where $\tau$ is a pre-defined temperature. Notably, with the presence of $C_c^-$, $C_S$ from Eq. (5) will be re-formulated as $C_S = C_S \cup C_c^-$, which is equivalent to increasing the number of wrong answers.

E. Loss function

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

$$L = L_{TM} + \lambda_{CR} L_{CR},$$

(7)

where $\lambda_{CR}$ is a penalty term. Notably, there are another two training strategies, including pre-train and alternate. The former is to update the model first using $L_{TM}$ before fine-tuning with $L_{CR}$, while the latter is to train the model with $L_{TM}$ for $(N_t - 1)$ iterations and switch to $L_{CR}$ once, for every $N_t$ iterations. However, the experimental results show the joint training outperforms pre-train and alternate based model.

F. Discussion

The relationship between the proposed method and existing pairwise algorithms is analysed in this section. Previous studies measure the matching representation (i.e., $C$ from Eq. (5)) using the following estimation:

- CNN-Matching [11]:
  $$H_q^a = \text{Enc}(q; a); H_p = \text{Enc}(p);$$
  $$M^{pp} = \text{Att}(H_q^a, H_p^a); M^{ap} = \text{Att}(H_q^a, H_p^p);$$
  $$C = \text{Sim}(H_q^a, M).$$

- Co-Matching [12]:
  $$H_q^a = \text{Enc}(q); H_p^a = \text{Enc}(a); H_p^p = \text{Enc}(p);$$
  $$M^{pp} = \text{Att}(H_q^a, H_p^p); M^{ap} = \text{Att}(H_q^a, H_p^p);$$
  $$C = [\text{Sim}(M^{pp}, H_p^p); \text{Sim}(M^{pp}, H_p^p)].$$

- DCMN+ [7]:
  $$H_q^a = \text{Enc}(q); H_p^a = \text{Enc}(a); H_p^p = \text{Enc}(p);$$
  $$M^{qa} = \text{Att}(H_q^a, H_p^a); M^{pa} = \text{Att}(H_q^a, H_p^p);$$
  $$M^{pp} = \text{Att}(H_q^p, H_p^p);$$
  $$C = [\text{Gat}(M^{qa}, M^{qa}); \text{Gat}(M^{qa}, M^{pa});$$
  $$\text{Gat}(M^{pa}, M^{pa})].$$

where $\text{Enc}$ represents the encoder, $\text{Att}$ stands for the attention operation, $\text{Sim}$ is for the similarity calculation, $\text{Gat}$ is a rest gate function, and $[;]$ is the vector concatenation. Note that existing methods adopted different implementation of $\text{Enc}$, $\text{Att}$, and $\text{Sim}$, etc. For instance, $\text{Enc}$ in [11] and [12] has been implemented as CNN and BERT, respectively.

Compared to these 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:

$$M^{qa} = \text{Att}(H_q^a, H_p^a); M^{pa} = \text{Att}(H_q^a, H_p^p);$$
$$M^{pp} = \text{Att}(\text{Att}(M^{qa}, M^{pa}), H_p^a);$$
$$M^{pp} = \text{Att}(\text{Att}(M^{qa}, M^{pa}), H_p^a);$$

(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.

IV. EXPERIMENTS

A. Datasets

Two datasets adopted in the experiments are RACE [15] and DREAM [19]. RACE is one of the widely used and largest benchmark 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 and three options associated with each individual question. Their statistics are shown in Table II.

B. Implementation and settings

Four pre-trained language models, including the BERT-base and BERT-large [8], RoBERTa-large [9], and Albert-xxlarge [10] are adopted as the word-embedding encoders. Their detail settings can be found in Table III.

The dropout rate for four encoders is set as 0.1, while the Adam optimizer is adopted to fine-tune the CTM model. 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 [20] 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 $\tau = 0.07$. All experiments are completed on a machine with four Tesla K80 GPUs. For document preprocessing, we use NLTK to remove stop words and symbols. 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 ($n_q$).

### C. Results

The performance of the proposed CTM is compared with several existing methods, including

- traditional pre-trained models, such as BERTbase/\textit{large} [8], RoBerta-large [9], and Albert-xxlarge [10], that leverage the representation of the special token ([CLS]) as the overall matching of the input triple;
- DCMN+ [7], an algorithm that proposes dual co-matching network to model the relationship among passage, question and answer bidirectionally;
- CSFN [5], an algorithm that utilizes the semantic role labeling to strengthen the correlation representation;
- DUMA [14], an algorithm that generalizes the architecture of Multi-head Attention to the triple-representation scenario;
- ConceptPlug [6], an algorithm that employs external knowledge graph(s) to fine-tune the MCQA task;
- MMM [20], an algorithm that applies a transfer-learning strategy to first coarse-tune the model using several out-of-domain datasets before fine-tuning with a large in-domain dataset;
- SG-Net [13], an algorithm that combines syntactic tree information with the pre-trained encoder for linguistic matching.

Results of the proposed CTM and compared methods are shown in Table IV. The proposed CTM substantially outperforms existing methods, via achieving state-of-the-art performance on both RACE and DREAM datasets.

Not surprisingly, the BERT-base methods achieve generally worse performance compared to their large-scale counterparts (such as BERT-large or Albert-xxlarge), which shows the benefit from a better pre-trained model. Although the baseline performance can be further improved by bidirectional matching [7] and/or external knowledge [13] (using the same encoder), these strategies only consider pairwise matching among the passage, question and answer independently. As such, their results are notably worse than the proposed method.

It is also worth noting that CTM works robustly with various encoders, that boosts up the performance to varying extents. In general, the weaker encoder is, the larger performance boost CTM achieves (except for the BERT-large encoder with the RACE-M dataset). Taking DREAM as an example, the accuracy from the BERT-base + CTM model improves that of BERT-base by 6 percentage points, compared to that of Albert-xxlarge (2.2 percentage points). The main reason comes from the adopted encoders themselves, as the encoding capability and the scale of backbone parameters are different. On the other hand, existing methods (such as DCMN+ and MMM) are more backbone-dependent. For instance, MMM only improves 0.1 percentage points with BERT-large for the RACE-M dataset, which is significantly lower than the case of BERT-base (2.9 percentage points). Similarly, DCMN+ achieves even worse performance (with RoBerta-large (86.1) and Albert-xxlarge (88.5)) compared to the original model (86.5 and 89.0) for RACE-M. By contrast, CTM is more backbone-free, which indicates that its robustness towards various encoders (pre-trained models).

### D. Ablation study

To begin with, experiments are conducted on the RACE-H dataset (using the BERT-base encoder) to validate the individual contribution from the proposed TM and CR modules to the final performance.

Results from Fig 2 evidently states their effectiveness. First, either the individual TM or CR module improves the model accuracy compared to the baseline (BERT-base). Second, the TM module brings the larger performance boost compared to CR. The result highlights the significance of forming the component correlation from the triple as a whole. On the other hand, the purpose of CR is to separate correct and wrong triples, so it might not be directly useful to identify correct answers. As such, the accuracy from the CR-only model is slightly lower than that of TM. At last, combing both TM and CR modules achieves the best outcome (shown as the

---

**Table II**

**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   |

**Table III**

**Settings for adopted encoders, where HS is the hidden size, Layer is the number of hidden layer, Heads is the number of self-attention heads, and L.rate represents the adopted learning rate, respectively.**

| Model     | HS | Layer | Heads | L.rate  |
|-----------|----|-------|-------|---------|
| BERT-base | 768| 12    | 12    | 2e-5    |
| BERT-large| 1024| 24    | 16    | 2e-5    |
| RoBerta-large | 1024| 24    | 16    | 2e-5    |
| Albert-xxlarge | 4096| 12    | 64    | 1e-5    |
Table IV
RESULTS IN ACCURACY (%), OBTAINED BY CTM AND THE COMPARING METHODS ON THE ON THE TEST SET. “×” 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-base       | 71.1   | 62.3   | 63.2  |
| + DCMN+         | 73.2   | 64.2   | ×     |
| + CSFN★         | 68.3   | 68.3   | 64.0  |
| + DUMA          | ×      | ×      | 64.0  |
| + ConceptPlug★  | 65.3   | 65.3   | 65.3  |
| + MMM           | 74.8   | 65.2   | ×     |
| + CTM           | 75.2   | 68.3   | 69.2  |
| BERT-large      | 76.6   | 70.1   | 66.8  |
| + DCMN+         | 79.3   | 74.4   | ×     |
| + ConceptPlug★  | 72.6   | 72.6   | 69.3  |
| + MMM           | 78.1   | 70.2   | ×     |
| + CTM           | 81.5   | 75.3   | 72.0  |
| RoBerta-large   | 86.5   | 81.3   | 85.0  |
| + DCMN+         | 89.1   | 83.3   | 88.9  |
| + CTM           | 90.2   | 84.1   | 89.4  |
| Albert-xxlarge  | 89.0   | 85.5   | 88.5  |
| + DCMN+         | 88.5   | 84.6   | 87.8  |
| + DUMA          | 90.9   | 86.7   | 90.4  |
| + SG-Net        | 89.2   | 86.1   | ×     |
| + CTM           | 91.4   | 87.8   | 90.7  |

Figure 2. Ablation study on TM and CR modules from the proposed CTM for the RACE-H dataset. For the "Both" case, we further consider scenarios with different $\lambda_{CR}$ values as $\{0, 0.5, 1, 1.5\}$.

"Both" case, which again demonstrates the superiority of the proposed triple matching and capturing semantic differences among answer candidates.

The penalty term $\lambda_{CR}$ from Eq. (7) is further verified, that is used to balance the TM and CR module in the "Both" case. Specifically, a range of $\lambda_{CR}$ (i.e., $\{0, 0.5, 1, 1.5\}$) are considered. Notably, in the case of $\lambda_{CR} = 0$, the model degrades to the TM(only) module. The comparison results, presented in the zoom-in area from Fig. 2, reveal that the proposed CR helps, with the presence of TM, in enhancing the matching capability (via maximizing the feature difference between the correct and wrong triple). Specifically, CTM achieves the best result (68.3) when $\lambda_{CR} = 0.5$, compared to that of $\lambda_{CR} = 0$ (65.7, the TM(only) module). Yet, the increase of the $\lambda_{CR}$ value results in the inferior accuracy (in particular with $\lambda_{CR} = 1.5$). The reason, again, could be the compatibility between the learned features and the final classification. With a larger $\lambda_{CR}$, the model tends to learn distinct features to separate answers, which might not be useful to select the correct answer. As a consequence, the setting of $\lambda_{CR} = 0.5$ is adopted in the following.

Next, we further examine Triple Matching and Contrastive Regularization module to manifest their efficacy.

On Triple Matching This experiment aims to compare the performance of two different matching strategies, i.e. the proposed TM against existing pairwise one. Accordingly, the contrastive regularization in this experiment is disabled by setting $\lambda_{CR} = 0$.

The DCMN+ model [7] is adopted as the pairwise-based opponent, which achieves the state-of-the-art performance within previous work. It consists of three dual-matching branches: question-answer pair ($M^{qa}$), question-passage pair ($M^{qp}$), and answer-passage pair ($M^{ap}$). By contrast, the proposed CTM includes three branches, including answer-aware context matching $M^a$, passage-aware context matching $M^p$, and question-aware context matching $M^q$, respectively. We then carefully ablate those branches by enumerating different combinations, and compare their performance with that of DCMN+.

Table V
THE PERFORMANCE COMPARISON BETWEEN THE PROPOSED CTM AND DCMN+ ON THE RACE-H TEST SET, BY TAKING DIFFERENT MATCHING BRANCHES INTO ACCOUNT.

| Branch                        | Acc   | Branch                        | Acc   |
|-------------------------------|-------|-------------------------------|-------|
| $M^{qp}; M^{ap}$              | 63.8  | $M^{qp}; M^{ap}; M^{qa}$      | 63.3  |
| $M^{qp}; M^{ap}$              | 62.1  | $M^{qp}; M^{ap}; M^{qa}; M^{pp}$ | 64.2  |
| $M^{q}$                       | 48.6  | $M^{p}$                       | 64.1  |
| $M^{p}$                       | 63.8  | $M^{p}; M^{q}$                | 64.6  |
| $M^{q}; M^{p}$                | 64.8  | $M^{p}; M^{q}; M^{p}$         | 52.3  |
| $M^{p}; M^{q}; M^{p}$         | 65.7  |                               |       |

Table V shows the results on the proposed TM and DCMN+. As observed, the branch of $M^q$ contributes mostly in the answer selection, as it achieves the highest accuracy among proposed branches. On the other hand, the single $M^q$ branch obtains the worst performance. Note that the $M^q$ is to take the answer as the prior context for the subsequent matching. Yet, a single answer itself fails to provide a unique information (for instance a wrong answer might become correct with another question), so its matching contribution is less significant. This result suggests the importance of utilizing question(s) as the context, rather than passage and/or answers, for the MCQA tasks.
In addition, the combination of all three branches achieves the best matching outcome (65.7%) compared to that of DCMN+ (64.2%). The result not only indicates the necessity of utilizing all three proposed matching branches, but also shows the superiority of the triple matching compared to existing dual matching.

**On Contrastive Regularization** Fixing $\lambda_{CR} = 0.5$, we further perform the ablation study on the CR module to manifest the efficacy from positives ($C_+^c$) and/or negatives ($C_-^c$) from Eq. (6). Specifically, we consider the following two variants of the CR module:

- the “+p” case only considers to add one positive $C_+^c$, while original wrong answers are cast as negatives without creating additional ones (i.e., $C_+^- = \emptyset$; and
- the “+n” differs from “+p” by creating only one extra negative ($C_-^c$, and no positives $C_+^c = \emptyset$). Due to the absence of $C_+^c$, Eq. (6) is disabled. However, $C_S$ from Eq. (5) is reformulated as $C_S = C_S \cup C_-^c$ which is equivalent to adding one additional wrong answer;

At last, we use “+both” to indicate the proposed CR loss, where both the positive and negative are involved. To make a reference, both the BERT-base and DCMN+ models are also considered to verify the above variants.

![Figure 3. Performance comparison in terms of variants of the CR loss, where the baseline result represents the normal matching model.](image)

The results from Fig 3 summarized contributions of adding positives and/or negatives to the final performance. As observed, the proposed CR module (including its variants) is agnostic to the employed encoders and matching methods, which appropriately learns informative representations to improve the model accuracy. All three employed models have been observed a performance boost with the help of CR. For instance, adding $C_+^c$ (in relation to the case of “+p”) is in favor of the model via awarding the answer difference (or pushing away wrong passage-question-answer triples from the correct ones). Similarly, adding extra negative sample (associated with the case of “+n”) enforces the model to learn a better representation for the correct triple, due to the increasing number of wrong answers. Consequently, either “+p” or “+n” leads to an improved result, while the averaged accuracy has been improved 1.3, 0.6, and 1.7 for BERT, DCMN+ and CTM respectively.

On the other hand, the combination of adding $C_+^c$ and $C_-^c$ outperforms other model variants, that evidently states the effectiveness of the proposed CR loss. Notably, compared to BERT or DCMN+ with(out) CR, the TM+CR works best and achieves a 2.6 (biggest) performance boost. The result also indicates the superiority of TM via extracting more-advanced features than that of BERT and DCMN+.

**Case analysis** In this section, the model capability is further analyzed based on the question category (in terms of the question type and the number of supporting sentences) and answer similarity (in terms of the number of duplicated words across candidate answers). Followed by [7], we randomly select approximately 10% samples (350 questions) from the test set of RACE-H, and the pairwise matching method (i.e., DCMN+) is again employed for the comparison purpose.

To begin with, selected samples are manually annotated using question types of what, which, close and other. The “other” type including questions with why, who, when, where, and how. Additionally, they are further tagged based on the number of sentences (#s) required to answer those questions. The performance from two models is accordingly shown in Table VI.

![Table VI. COMPARISON OF THE PROPOSED CTM AND DCMN+ 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+.](image)

| Type   | #s=1  | #s=2  | #s≥ 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) |
| close  | 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 indicates the superiority of the proposed algorithm when answering questions such as cloze tests and with more supporting sentences involved. Specifically, the cloze test requires the reasoning capability from the model to scan the entire passage according to the given question and estimate the matching/fitness for all candidate answers. As such, the proposed triple matching is more suitable than the conventional pairwise 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 differences, thereby achieving the improvement for the question answering.

Similarly, for questions that need to infer from (more than) 3 sentences, the result also 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 stably than its
counterpart, which shows its robustness of handling cases with multiple evidence sentences.

On the other hand, the model performance against the answer similarity is shown in Table VII. The result shows that CTM outperforms DCMN+ in all scenarios, in particular with cases of the answer similarity for $\geq 3$. Note that the average answer length is 6.8 (shown in Table II). The comparison clearly demonstrates that the proposed method is able to identify correct answers, even if candidates contain approximately 50% lexically same words. By contrast, the pairwise opponent has been largely impacted by the answer similarity, thereby obtaining a much worse accuracy.

| Model      | 0    | 1    | 2    | $\geq 3$ | All  |
|------------|------|------|------|---------|------|
| DCMN+      | 30.6 | 31.7 | 19.1 | 18.6    | 100.0|
| CTM        | 67.7 | 64.2 | 62.5 | 60.1    | 64.2 |
|            | 69.5 | 68.1 | 67.8 | 66.9    | 68.2 |

In conclusion, it can be empirically confirmed that the proposed CTM algorithm achieves comparative performance than pairwise methods, in particular for questions requiring more supporting sentences and big answer similarity.

V. 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 contrastive regularization is further introduced to encourage the

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

[1] Y. Seonwoo, J.-H. Kim, J.-W. Ha, and A. Oh, “Context-aware answer extraction in question answering,” in Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Nov. 2020, pp. 2418–2428.
[2] M. Joshi, D. Chen, Y. Liu, D. S. Weld, L. Zettlemoyer, and O. Levy, “SpanBERT: Improving pre-training by representing and predicting spans,” Transactions of the Association for Computational Linguistics, vol. 8, pp. 64–77, 01 2020.
[3] H. Sun, B. Dhingra, M. Zaheer, K. Mazaitis, R. Salakhutdinov, and W. W. Cohen, “Open domain question answering using early fusion of Knowledge Bases and text,” in Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, 2018.
[4] J. Liu, S. Mei, X. Hu, X. Yao, J. Yang, and Y. Guo, “Seeing the wood for the trees: a contrastive regularization method for the low-resource knowledge base question answering,” in Findings of the Association for Computational Linguistics: NAACL 2022. Seattle, United States: Association for Computational Linguistics, Jul. 2022, pp. 1085–1094.
[5] Q. Duan, J. Huang, and H. Wu, “Contextual and semantic fusion network for multiple-choice reading comprehension,” IEEE Access, vol. 9, pp. 51 669–51 678, 2021.
[6] R. Li, Z. Jiang, L. Wang, X. Lu, M. Zhao, and D. Chen, “Enhancing Transformer-based language models with commonsense representations for knowledge-driven machine comprehension,” Knowledge-Based Systems, vol. 220, p. 106936, 2021.
[7] S. Zhang, H. Zhao, Y. Wu, Z. Zhang, X. Zhou, and X. Zhou, “DCMN+: Dual co-matching network for multi-choice reading comprehension,” Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, no. 05, pp. 9563–9570, 2020.
[8] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “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, Jun. 2019, pp. 4171–4186.
[9] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, “Roberta: A robustly optimized BERT pretraining approach,” CoRR, vol. abs/1907.11692, 2019.
[10] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Socher, “Albert: A lite bert for self-supervised learning of language representations,” arXiv, 2019.
[11] A. Chaturvedi, O. Pandit, and U. Gazain, “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), Jul. 2018, pp. 272–277.
[12] S. Wang, M. Yu, J. Jiang, and S. Chang, “A co-matching model for multi-choice reading comprehension,” in Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, Jul. 2018, pp. 746–751.
[13] Z. Zhang, Y. Wu, J. Zhou, S. Duan, H. Zhao, and R. Wang, “SG-Net: Syntax-guided machine reading comprehension,” in Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence, vol. 34, no. 5, 2020, pp. 9636–9643.
[14] P. Zhu, Z. Zhang, H. Zhao, and X. Li, “DUMA: Reading comprehension with transposition thinking,” IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 30, pp. 269–279, 2022.
[15] G. Lai, Q. Xie, H. Liu, Y. Yang, and E. Hovy, “RACE: Large-scale ReAding comprehension dataset from examinations,” in Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Sep. 2017, pp. 785–794.
[16] H. Zhu, F. Wei, B. Qin, and T. Liu, “Hierarchical attention flow for multiple-choice reading comprehension,” Proceedings of the AAAI Conference on Artificial Intelligence, vol. 32, no. 1, pp. 6077–6084, Apr. 2018.
[17] T. Gao, X. Yao, and D. Chen, “SimCSE: Simple contrastive learning of sentence embeddings,” arXiv preprint arXiv:2104.08021, 2021.
[18] X. Liang, L. Wu, J. Li, Y. Wang, Q. Meng, T. Qin, W. Chen, M. Zhang, and T.-Y. Liu, “R-Drop: Regularized dropout for neural networks,” arXiv preprint arXiv:2106.14448, 2021.
[19] K. Sun, D. Yu, J. Chen, D. Yu, Y. Choi, and C. Cardie, “DREAM: A challenge dataset and models for dialogue-based reading comprehension,” Transactions of the Association for Computational Linguistics, vol. 7, pp. 217–231, 04 2019.
[20] D. Jin, S. Gao, J.-Y. Kao, T. Chung, and D. Hakkani-tur, “MMM: Multi-stage multi-task learning for multi-choice reading comprehension,” Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, no. 05, pp. 8010–8017, Apr. 2020.