Multi-Matching Network for Multiple Choice Reading Comprehension

Min Tang and **Jiaran Cai** and **Hankz Hankui Zhuo***
School of Data and Computer Science, Sun Yat-Sen University, Guangzhou, China.
{tangm28@mail2,zhuohank@mail}.sysu.edu.cn

Abstract

Multiple-choice machine reading comprehension is an important and challenging task where the machine is required to select the correct answer from a set of candidate answers given passage and question. Existing approaches either match extracted evidence with candidate answers shallowly or model passage, question and candidate answers with a single paradigm of matching. In this paper, we propose **M**ulti-**M**atching **N**etwork (**MMN**) which models the semantic relationship among passage, question and candidate answers from multiple different paradigms of matching. In our **MMN** model, each paradigm is inspired by how human think and designed under a unified compose-match framework. To demonstrate the effectiveness of our model, we evaluate **MMN** on a large-scale multiple choice machine reading comprehension dataset (i.e. RACE). Empirical results show that our proposed model achieves a significant improvement compared to strong baselines and obtains state-of-the-art results.

Introduction

As a fundamental task and a long-standing goal in the field of natural language processing, machine reading comprehension (MRC) aims to enable machines to automatically answer questions according to passages in hand. There have been many researches on machine reading comprehension. For example, (Yin et al. 2016) proposed to match passages against sequences that concatenate both questions and candidate answers; (Dhingra et al. 2017; Chen, Bolton, and Manning 2016; Lai et al. 2017; Zhu et al. 2018) proposed to first match passages to questions and then select answers based on the matching result; etc. Despite the success of previous approaches on reading-comprehension scenarios that answers can be directly extracted from the given passages, such as SQuAD (Rajpurkar et al. 2016) and CNN/Daily Mail (Hermann et al. 2015), they do not work on questions whose answers need to be inferred from the given questions and passages, i.e., answers cannot be directly extracted from passages. One example of such reading-comprehension scenarios is RACE, which was recently released by (Lai et al. 2017). RACE was built from middle and high school English examinations in China. As mentioned in (Lai et al. 2017), RACE is more challenging and requires more inferences compared to the above-mentioned datasets.

For example, in Table 1, we can see that there is contradiction between "Mike ***never*** *washed them well" in the passage and combination of "Mike" in Question 1 and "washed them clean" in candidate answer* ***A****; there is entailment between "Mike never washed them well" in the passage and combination of "Mike" in Question 1 and "never washed them clean" in candidate answer* ***C****.*

To address this problem, (Wang et al. 2018) propose to jointly model the sequence triplets (i.e. passage, question and candidate answer) assuming that questions and candidate answers are equally important in reading comprehension. Triplet matching, however, usually encodes the locational information of the question and the candidate answer matched to a specific context of the passage (Wang et al. 2018), which ignores scenarios that there are multiple evidence snippets in the passage, which are significant for answering the questions. For example, in Question 2, "the main idea of this passage" depends on the evidence snippets as described by all sentences in the passages, which produces the answer "D", i.e., "The job market has changed dramatically over the past 4 years". In this paper, we aim to build a novel framework to capture multiple evidence snippets and entailment relationships among passages, questions and answers, which is challenging since we cannot find answers by locally matching words among passages, questions and answers.

To overcome the challenge, we observe that humans usually answer multiple choice questions by two ways. The first one is to extract evidence snippets from passages according to questions, and then match evidence snippets with candidate answers. The other way is to read candidate answers and questions together to form pseudo statements and then recognise entailment relationships between pseudo statements and passages. After that, human fuse different considerations to verify answers and make final decisions.

Inspired by how humans answer multiple choice questions, we propose a novel approach, called **MMN**, which stands for **M**ulti-**M**atching **N**etwork. Corresponding to above-mentioned two ways, our **MMN** approach contains two types of matching between multiple sequences, namely, (1) Evidence-Answer Matching and (2) Question-Passage-Answer Matching, which are designed under the unified compose-match framework. We first encode the context in-

*corresponding author

Table 1: Examples in RACE. The text in bold is the supportive evidence or related premise to answer the questions.

Passage: $\cdots$ Three or four times every day his mother said to him," Mike, your hands are very dirty again, go and wash them." **Mike never really washed them well. He only put his hands in the water for a few seconds and then took them out again.** $\cdots$	
Question 1:When Mike washed his hands, _.	**Golden answer**: C
A. He washed them clean. **B**. He used soap and water.	**C**. He never washed them clean. **D**. He felt very happy.
Passage: $\cdots$ **This year's college graduates are facing one of the worst job markets in years.** $\cdots$ **many people already working are getting laid off and don't have jobs, so it's even harder for new college graduates to find jobs**. $\cdots$ **Other popular fields (like information system management, computer science, and political science) have seen big declines in starting salaries**. $\cdots$ **which would be great in bad economy.**" $\cdots$	
Question 2: The main idea of this passage is that _?	**Golden answer**: D
A. A lot of graduates are losing their jobs. **B**. Ryan Stewart has not been able to find a job.	**C**. Salaries in some fields have increased in the past year. **D**. The job market has changed dramatically over the past 4 years.

formation into word embeddings and generate the contextualized word representations with GRUs (Chung et al. 2014) and gate mechanism. After that, we develop Evidence-Answer Matching to extract multiple evidence snippets to form evidence sequences, which are then matched to candidate answers. Meanwhile, we build Question-Passage-Answer Matching to learn semantic relationships among passages, questions and candidate answers. Finally, we integrate multiple aggregated matching results as final matching representations to make final decisions.

The remainder of this paper is organized as follows. We first describe our **MMN** approach in detail in Section 2 and present our experimental results in Section 3. After that, we present previous work related to our work in Section 4 and conclude our work with future work in Section 5.

Our Proposed Model

In this section, we introduce the task definition and describe our **MMN** model in detail. The illustration of **MMN** is shown in Figure 1(a). Our model is composed of following major components: input embedding, projection layer, context encoding, multi-matching component, merging layer and answer prediction. We will address each component in detail in the following subsections.

Task Definition

In the scenario of multiple choice reading comprehension, given a passage, a question and a few candidate answers, our goal is to select the correct answer from candidate answers. Formally, we represent the dataset as $\{P, Q, \mathbb{A}, y\}_{i=1}^{N}$, where $P = \{w_t^P\}_{t=1}^{l_p}$ is a passage composed of a sequence of words w_t^P, $Q = \{w_t^Q\}_{t=1}^{l_q}$ is a question composed of a sequence of words w_t^Q, $\mathbb{A}$ is a set of answers, each of which is $A = \{w_t^A\}_{t=1}^{l_a} \in \mathbb{A}$. l_p, l_q, l_a are lengths of the passage, question and answer, respectively. In the sequel, for the simplicity of description, we will omit the superscript (P, Q or A) of w^P, w^Q and w^A, and the subscript (p, q, or a) of l_p, l_q and l_a.

Input Embedding + Projection Layer

The goal of input embedding is to map one-hot encoded word vectors into low dimensional vector space. The output of input embedding consists of three parts: pretrained word-level word embedding vector, char-level word embedding vector and exact word matching feature. Following (Seo et al. 2016), char-level word embeddings are generated by applying convolution and max-over-time-pooling operation to each word. Then, we pass the the output $e \in \mathbb{R}^{n \times l}$ of input embedding into a projection layer to learn task-specific representation $E \in \mathbb{R}^{d \times l}$ as follows:

$$E = ReLU(W^P e + b^P), \tag{1}$$

where $W^P \in \mathbb{R}^{d \times n}, b^P \in \mathbb{R}^d$ are weights and biases, $ReLU$ is the Rectified Linear Unit, n denotes the number of dimensions of input embedding vector and d denotes the number of hidden units in the projection layer. The projection layer outputs a sequence of d dimensional vectors for input sequences (i.e., passages, questions and answers).

Context Encoding

In order to accumulate contextual representations for words, we employ a bi-directional recurrent network (BiRNN) to read sequences from both sides, i.e. the bottom part in Figure 1(a). Specifically, we use Gated Recurrent Unit (GRU) (Chung et al. 2014) as the basic building block and concatenate the hidden states of both directions at each time step. We denote the operation of BiGRU on a sequence s as $BiGRU(s)$. Thus, we have contextualized word representations $H \in \mathbb{R}^{d \times l}$ as follows:

$$H = BiGRU(E). \tag{2}$$

Inspired by (Srivastava, Greff, and Schmidhuber 2015), we exploit gate mechanism to control the information flow from word representations and contextualized representations. Thus, we have gated contextualized representations $\tilde{H} \in \mathbb{R}^{d \times l}$ as shown below:

$$z = \sigma(W^E E + W^H H + b), \tag{3}$$

$$\tilde{H} = E * z + H * (1 - z), \tag{4}$$

where $W^E, W^H \in \mathbb{R}^{d \times d}, b \in \mathbb{R}^d$ are weights and biases, and $z \in \mathbb{R}^{d \times l}$ is the *reset gate*. Essentially, the gated contextual encoding layer can be seen as a sequential variant of the highway network, where we use a recurrent neural network to learn the gate instead of a forward neural network. Intuitively, incorporating sequential information into a highway network could be more effective in modelling sequences.

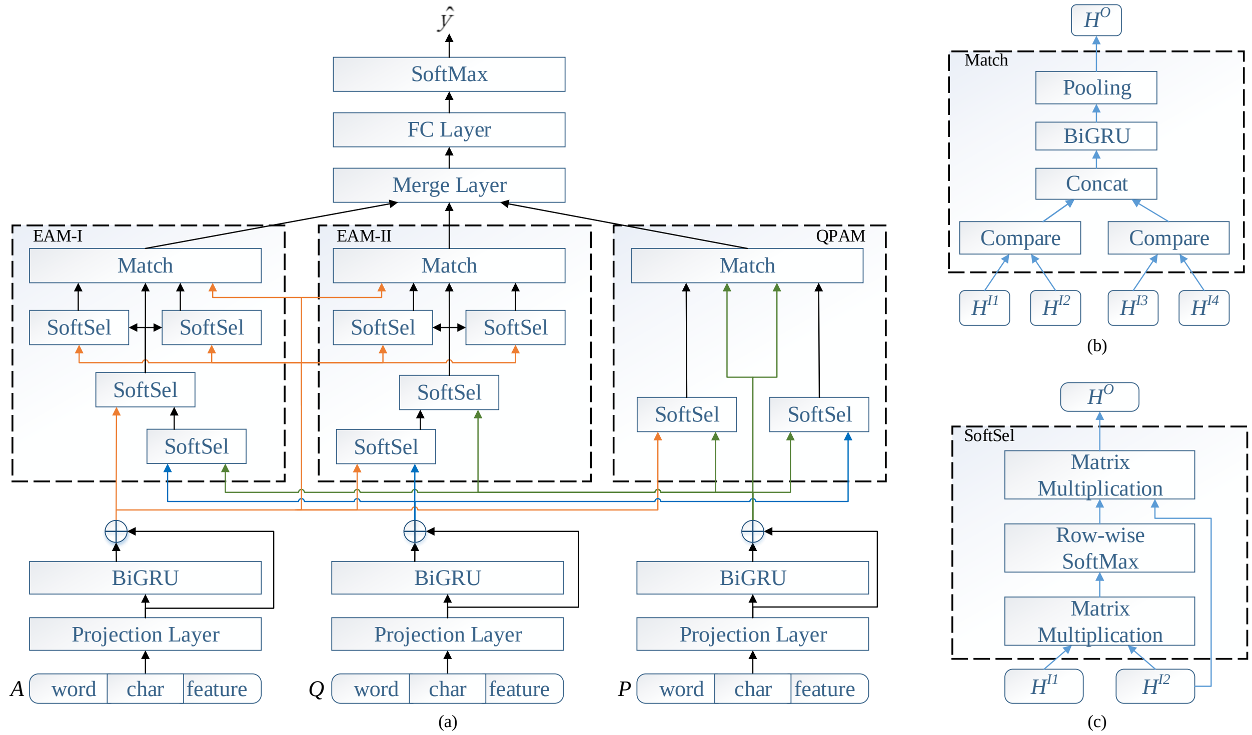

Figure 1: (a) Illustration of **MMN** (Left: Evidence-Answer Matching I; Middle: Evidence-Answer-Matching II; Right: Question-Passage-Answer Matching) (b) Match operation. (c) SoftSel operation.

Multi-Matching Component

In the multi-matching component, there are two types of matching modules, i.e. Evidence-Answer Matching module (denoted by EAM-I and EAM-II in Figure 1(a)) and Question-Passage-Answer Matching module (denoted by QPAM in Figure 1), which are constructed under a unified framework. Both of them consist of two submodule: *composing submodule* and *matching submodule*. Composing submodule is used to build attended sequences according to the attention mechanism, while matching submodule is used to model the semantic relationship between sequences. Before we describe them in detail, we define two operations, **SoftSel** and **Match** as follows.

- **SoftSel** (i.e. Figure 1(c)): This operation takes two sequences as input, and outputs a sequence. After calculating the cartesian similarity between all possible combinations of vectors of the two input sequences, we apply a row-wise softmax function to the similarity, and then we calculate the weighted sum vector of the first sequence at each position of the second sequence. Let $H^{I_1} \in \mathbb{R}^{d \times l_1}$ and $H^{I_2} \in \mathbb{R}^{d \times l_2}$ be the input. We calculate the output $H^O \in \mathbb{R}^{d \times l_1}$ as follows:

$$G = {H^{I_1}}^T W^G H^{I_2}, \tag{5}$$

$$\tilde{G} = row\text{-}wise\ softmax(G), \tag{6}$$

$$H^O = H^{I_2} \tilde{G}^T, \tag{7}$$

  where $W \in \mathbb{R}^{d \times d}$ are weights, $G, \tilde{G} \in \mathbb{R}^{l_1 \times l_2}$ are immediate similarity matrices. Intuitively, H^O_i encodes the most relevant part of the second sequence w.r.t. ith word in the first sequence.

- **Match** (i.e. Figure 1(b)): This operation takes four sequences as input, which are denoted by $H^{I_1}, H^{I_2}, H^{I_3}, H^{I_4} \in \mathbb{R}^{d \times l}$, respectively, and outputs the aggregated matching representation $H^f \in \mathbb{R}^{2d}$. To do this, we first divide four inputs into two groups, i.e., the first two as one group and the last two as the other group. We then calculate the matching representation $M^1, M^2 \in \mathbb{R}^{d \times l}$ within each group by:

$$M^1 = ReLU(W^M \begin{bmatrix} H^{I_2} * H^{I_2} \\ H^{I_1} - H^{I_2} \end{bmatrix} + b^M), \tag{8}$$

$$M^2 = ReLU(W^M \begin{bmatrix} H^{I_3} * H^{I_4} \\ H^{I_3} - H^{I_4} \end{bmatrix} + b^M), \tag{9}$$

where $W^M \in \mathbb{R}^{d \times 2d}$ and $b^M \in \mathbb{R}^d$ are learnable weights and biases and $\begin{bmatrix} \cdot \\ \cdot \end{bmatrix}$ denotes column-wise concatenation, $*$ and $-$ denote element-wise multiplication and substraction, respectively. After that, we project the concatenation of M^1 and M^2 to d dimension space and aggregate the projected matching information $H^M \in \mathbb{R}^{d \times l}$ with a BiGRU layer as follows:

$$H^M = BiGRU(W^{H^M} \begin{bmatrix} M^1 \\ M^2 \end{bmatrix} + b^{H^M}), \tag{10}$$

where $W^{H^M} \in \mathbb{R}^{d \times 2d}$ and $b^{H^M} \in \mathbb{R}^d$ are learnable weights and biases. Finally, we extract the salient feature with max pooling and attentive pooling operation over aggregating representation to obtain $H^{max}, H^{att} \in \mathbb{R}^d$ and concatenate them as final output $H^f \in \mathbb{R}^{2d}$ as follows:

$$H^{max} = max\text{-}pooling(H^M), \tag{11}$$

$$\alpha = softmax(ReLU(W^{H^M} H^M + b^{H^M})), \tag{12}$$

$$H^{att} = H^M \alpha, \tag{13}$$

$$H^f = \begin{bmatrix} H^{max} \\ H^{att} \end{bmatrix}, \tag{14}$$

where $W^{H^M} \in \mathbb{R}^d$ and $b^{H^M} \in \mathbb{R}$ are learnable weights and biases and $\alpha \in \mathbb{R}^l$ are normalized parameters.

Evidence-Answer Matching Evidence-Answer Matching module is depicted at left and middle part of Figure 1(a), which aims to form matching information between the extracted evidences and candidate answers. There are two ways to reach this. We describe one of them in detail. The second one is similar to the first one so that we describe it briefly. We denote the first Evidence-Answer Matching module as EAM-I and the second Evidence-Answer Matching module as EAM-II. Evidence-Answer Matching modules (EAMs) indicate both of them.

We first begin by forming evidence sequence according to passage and question. We achieve this by applying a **SoftSel** operation which takes passage and question as input, which is formulated as:

$$\tilde{H}^Q = SoftSel(H^Q, H^P). \tag{15}$$

From the definition of **SoftSel**, we can see that $\tilde{H}^Q$ has the same length as the question, where each position at $\tilde{H}^Q$ is a weighted sum of all time steps of passage representation. Note that $\tilde{H}^Q$ is not a continuous slice of H^P and each position of $\tilde{H}^Q$ can be seen as a synthesized evidence vector. Due to different length of synthesized evidence and answer, it may be not suitable to match the answer and evidence directly, because there is huge semantic gap between them. As such, we utilize the answer to refine evidence further.

$$\tilde{H}^{QA} = SoftSel(H^A, \tilde{H}^Q), \tag{16}$$

where $\tilde{H}^{QA} \in \mathbb{R}^{d \times l_a}$. Intuitively, the most related evidence to answer is extracted, which will benefit the matching with the candidate answers.

Directly matching them may not be effective, so we explore a deeper fashion to model the deeper relationship. We calculate the attended sequence w.r.t. each other as follows:

$$\overline{H}^{QA} = SoftSel(\tilde{H}^{QA}, H^A), \tag{17}$$

$$\overline{H}^{A} = SoftSel(H^A, \tilde{H}^{QA}), \tag{18}$$

Together with two input sequences, we feed them into **Match** operation to get final aggregated matching representation $H^{f_1} \in \mathbb{R}^{2d}$ as follows:

$$H^{f_1} = Match(\hat{H}_{QA}, \overline{H}^{QA}, H^A, \overline{H}^A). \tag{19}$$

For the other way to form evidence, we first attend question with candidate answer. Then, attended question is used to interact with the passage to obtain evidence with the same length as the candidate answers. The intuition behind it is that some words in question is more important for extracting evidence. Finally, we feed extracted evidence sequence and answer into matching submodule to obtain final aggregated matching representation $H^{f_2} \in \mathbb{R}^{2d}$.

Question-Passage-Answer Matching Differing from Evidence-Answer Matching module, the Question-Passage-Answer Matching module (i.e. right part of Figure 1(a)) aims to model passage, question and candidate answer together, which we call QPAM for short. Like Evidence-Answer Matching, Question-Passage-Answer Matching is also designed under our unified compose-match framework. Following our framework, we first compose passage-aware question $H^{PQ} \in \mathbb{R}^{d \times l_p}$ and passage-aware candidate answer $H^{PA} \in \mathbb{R}^{d \times l_p}$ with **SoftSel** operation, which is formulated as follows:

$$H^{PQ} = SoftSel(H^P, H^Q), \tag{20}$$

$$H^{PA} = SoftSel(H^P, H^A). \tag{21}$$

Each position of H^{PQ}, H^{PA} represents the most relevant part of the question and the candidate answers, respectively. Next, we match H^{PQ}, H^{PA} with H^P using **Match** operation to obtain the aggregated matching representation $H^{f_3} \in \mathbb{R}^{2d}$, which is formulated as follows:

$$H^{f_3} = Match(H^{PQ}, H^P, H^{PA}, H^P). \tag{22}$$

Intuitively, the question and answer are combined implicitly to match with specific snippet in the passage to decide its entailment relationship.

Merging Layer

In this section, we merge the output vectors of Evidence-Answer Matching module and Question-Passage-Answer Matching module. In practice, merging outputs from different modules is proved to be very effective. Empirically, we concatenate all output vectors to obtain the summary of aggregated information $H^f \in \mathbb{R}^{6d}$ as follows:

$$H^f = [H^{f_1}; H^{f_2}; H^{f_3}], \tag{23}$$

where $[\cdot;\cdot;\cdot]$ denotes row-wise concatenation operation.

Answer Prediction

The input to the final prediction layer is the output of merging layer. We pass it to a hidden layer which outputs $v \in \mathbb{R}^{2d}$ as follows:

$$v = ReLU(W^v H^f + b^v), \tag{24}$$

where $W^v \in \mathbb{R}^{2d \times 6d}$ and $b^v \in \mathbb{R}^{2d}$ are trainable weights and biases. Then we apply a softmax classification layer to obtain the probability distribution over classes $\hat{y}$ as follows:

$$\hat{y} = softmax(Wv + b), \tag{25}$$

where $W \in \mathbb{R}^{2d}, b \in \mathbb{R}$ are trainable weights and biases.

Optimization Objective

We perform optimizing our proposed model with multi-class cross-entropy loss with L_2 regularization. The objective function is given as

$$J(\theta) = -\sum_{i=1}^{N}\sum_{j=1}^{L} y_j^{(i)} \log \hat{y}_j^{(i)} + \lambda||\theta||_2, \qquad (26)$$

where J is the cost function, y_j is a L dimensional one-hot vector with ground truth being 1 and the others being 0, $\hat{y}_j$ is the output probability of j^{th} class, and N, L, θ, λ denote the number of examples, the number of candidate answer for each question, all trainable parameters, regularization coefficient respectively.

Experiments

To evaluate the effectiveness of our model, we conduct experiments on RACE (Lai et al. 2017) which is a large-scale multiple choice reading comprehension dataset. Our model achieves the state-of-the-art performance on this dataset.

Dataset and Implementation Details

RACE dataset consists of two subsets collected from English exam for middle and high school students, which we call RACE-M and RACE-H respectively by following (Lai et al. 2017). All questions and candidate answers in RACE are generated by human experts. There is only one correct candidate answer among 4 candidates for each question. We partition the train/dev/test sets in the same way as ((Lai et al. 2017)) does and use accuracy as the evaluation metric. Accuracy is calculated as follows: $accuracy = N^+/N$, where N^+ and N are the number of correct predictions and the total number of questions in test set. The statistics of RACE dataset are shown in Table 2.

Table 2: Statistics of dataset. #w/p, #w/q and #w/a represent the average length of passage, question and candidate answers respectively. #a/q is the number of candidate answers for each question.

Dataset	Train	Dev	Test	#w/p	#w/q	#w/a	#a/q
RACE-M	25421	1436	1436	249.9	10.1	4.9	4
RACE-H	62445	3451	3498	374.9	11.4	6.8	4
RACE	87866	4887	4934	342.9	11.0	6.3	4

We tokenize all sentences using SpaCy toolkit[1] and lowercase all tokens. Our model is implemented with TensorFlow[2] and all hyperparameters are tuned according to performance on the development set. We use 300D GloVe[3] word embeddings which remain fixed during training. Out-of-vocabulary words are initialized to zero vectors. Each BiGRU holds 1 layer and 100 hidden units for each direction. To alleviate overfitting, we apply dropout (Srivastava et al. 2014) to the input of every layer with the dropout rate set to 0.2. The model is updated using mini-batch stochastic gradient descent with batch size 32. We optimize our model using ADAM (Kingma and Ba 2014) optimizer with learning rate 0.0003, where gradients are clipped in L2-norm to no larger than 10. Regularization coefficient is set to 1e-7. Early stopping technique is adopted after 50 epochs.

[1]https://spacy.io/

[2]https://www.tensorflow.org/

[3]https://nlp.stanford.edu/projects/glove/

Comparison against Baselines

We compare our model with the following baselines.

- SAR (Chen, Bolton, and Manning 2016) is a simple but effective model that uses bilinear attention to obtain evidence representation and then compare it to candidate answers.
- GAR (Dhingra et al. 2017) applies multi-hop gated attention mechanism between passage and question to obtain question-aware evidence representation.
- ElimiNet (Parikh et al. 2018) tries to eliminate candidate answers in a multi-hop manner, which is built on top of gated attention layer(s).
- HAF (Zhu et al. 2018) employs hierarchical attention flow to extract evidence and also considers correlation among candidate answers.
- DFN (Xu et al. 2017) utilizes various matching function to model sequences and selects the best policy optimized by reinforcement learning technique.
- Hier-Co-Matching (Wang et al. 2018) proposes a new co-matching approach to jointly model whether a passage can match both a question and a candidate answer.
- BiAttention (MRU) (Tay, Tuan, and Hui 2018) adopts bidirectional attention to obtain matching representation among sequences encoded by Multi-range Reasoning Units (MRU).

Performance comparison against all baseline models are shown in Table 3. From Table 3, we can observe that **MMN** outperforms all baselines and achieves state-of-the-art accuracy, which verifies the efficacy of our model. More specifically, on RACE-M subset, **MMN** outperforms the currently most competitive models BiAttention (MRU) and Hier-Co-Matching by 3.4 percentages and 5.3 percentages, respectively. On RACE-H subset, **MMN** achieves higher accuracy by 4.7 percentages and 4.0 percentages than BiAttention (MRU) and Hier-Co-Matching. Overall, **MMN** achieves an accuracy of 54.7%, which demonstrates the effectiveness of **MMN**. For further comparison, we also report results of the ensemble model. Following (Xu et al. 2017) and (Tay, Tuan, and Hui 2018), we build an ensemble model of 9 single models, where all single models are initialized with different random seeds and hyperparameters. We observe that ensemble models also obtains a significant performance gain. We also report the performace of Amazon Turkers tested on a sampled subset of RACE and the percentage of the unambiguous question in a subset of the test set (i.e. Ceiling Performance). Note that there is still a huge performance gap between machine reading model and human, which indicates the great potential for future research.

Table 3: Experimental results on test set. Best machine model result is in boldface. $*$ indicates ensemble model.

Model	RACE-M	RACE-H	RACE
SAR	44.2	43.0	43.3
GA	41.9	43.4	42.9
ElimiNet	44.4	44.5	44.5
HAF	46.2	46.4	46.0
DFN	51.5	45.7	47.4
Hier-Co-Matching	55.8	48.2	50.4
BiAttention (MRU)	57.7	47.5	50.4
MMN(Our model)	**61.1**	**52.2**	**54.7**
GA+ElimiNet*	47.4	47.4	47.2
DFN*	55.6	49.4	51.2
BiAttention (MRU)*	60.2	50.3	53.3
MMN*(Our model)	**64.7**	**55.5**	**58.2**
Turkers	85.1	69.4	73.3
Ceiling Performance	95.4	94.2	94.5

Ablation Study

To evaluate the effectiveness of each component of **MMN**, we conduct ablation analysis on RACE. Table 4 shows the performance of our full model and all ablated models on the development set. In Table 4, we observe that all key components of **MMN** contribute to the model performance. Without char-level word embedding, performance decreases by 0.7 percentage. To fairly validate the effect of gate mechanism used in context encoding, we replace it with highway network on top of BiGRU layer. Decreasing performance shows that gated contextual encoding is more effective than the highway network in our model. The key contribution of this work is multi-matching component which models the input in different ways. In Table 4, we see that performance gets worse when ablating EAM-I and EAM-II than ablating either of them. This is not surprising because a variety of extracting evidence could benefit learning more effective matching representation in merging layer. By removing QPAM module, the model only achieves accuracy of 53.2% and 48.8% on RACE-M and RACE-H respectively. We believe the reason is that there are many fill-in-blank questions in which recognizing entailment would be more suitable for selecting the correct answer.

Table 4: Results of ablated model on development set. CWE: Char-level Word Embedding. GCE: Gated Context Encoding.

Model	RACE-M	Δ	RACE-H	Δ	RACE	Δ
Full-Model	63.8	-	54.7	-	57.4	-
w/o CWE	62.3	-1.5	54.2	-0.5	56.7	-0.7
w/o GCE	61.2	-2.6	52.3	-2.4	54.9	-2.5
w/o EAM-I	62.6	-1.2	53.8	-0.9	56.3	-1.1
w/o EAM-II	62.0	-1.8	53.1	-1.6	55.7	-1.7
w/o EAMs	59.4	-4.4	50.5	-3.8	53.1	-4.3
w/o QPAM	**53.2**	**-10.6**	**48.8**	**-5.9**	**50.1**	**-7.3**

Accuracy w.r.t. Question Types

We first divide all examples in development set of RACE into many categories according to question type. Question types are decided by respective words, such as what, where, why, who. Besides questions whose types are indicated by some certain words, there are many fill-in-blank questions (e.g. _ is the movie capital of the world.) and statement-justification checking questions (e.g. which of the following is not true?), which are also divided into *fill-in-blank* and *true/false* respectively.

Figure 2 shows how well our model performs when dealing with different types of questions. In this experiment, we are interested to see whether our model performs well or badly on some particular type of questions. We can see that the performances for "why" questions are higher than others. The length of candidate answers of "why" question is usually longer than other types of questions, such as "when" and "where" question, which could provide richer information when matched with passage or evidence. However, performance gap against other question types is not large, which indicates that our model has robust performance in different type of questions. Our model works poorly for the true/false questions. Because it is difficult to utilize the sequence matching to handle questions with negative words, where flipping the semantic polarity of the sentence is required.

Accuracy w.r.t. Answer Length

We next evaluate the performance of our model with regard to the answer length. Since the length of answers varies in a large range, we divide all examples into several groups according to an average length, i.e. the number of words, of 4 candidate answers. In Figure 3, we see that our model performs better on RACE-M than on RACE-H in almost all groups of questions with different lengths of answers. It is not surprising because RACE-M is collected from exams in middle school. Note that it is not obvious that our model performs better when answering question with shorter candidate answers. The reason we believe is that longer candidate answers may provide more information as our model is based on sequence matching.

Case Study

To demonstrate the effectiveness of each subcomponent of **MMN**, we design an experiment to analysize the outputs of **MMN** and ablated model at final softmax classification layer. We sample two illustrative cases, which are also shown in Section 1. Normalized logits and predicted answers of different models are shown in Table 5. It is intriguing to note that our model can handle them well.

From Table 5, we can observe that both of **MMN** and QPAM predict the answer correctly when answering the first question, while EAM predicts the answer wrongly, which indicates our multi-matching can make a difference when one of single matching makes an incorrect decision. For the second question, as we stated in Section 1, it is necessary to extract multiple evidences to summarize the passage so that EAM is more suitable than QPAM to answer the question. We can see that both of **MMN** and EAM predict the

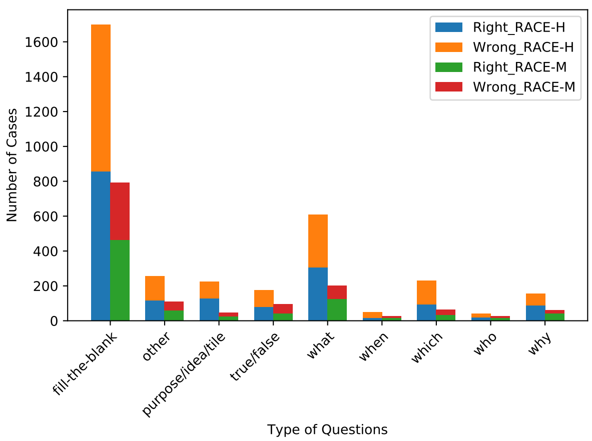

Figure 2: performance on different types of questions.

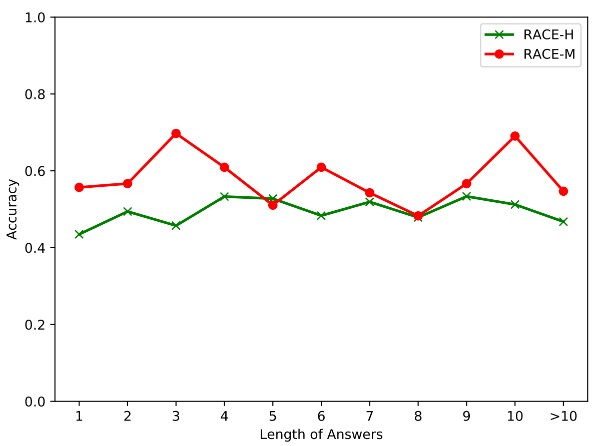

Figure 3: Performance on different lengths of answers.

answer correctly and QPAM predict the answer wrongly. Furthermore, It can be observerd that the output logit corresponding correct answer of **MMN** is higher than both EAM and QPAM for both two question. Intuitively, **MMN** benefits learning more rich representation in merging layer.

Table 5: Output logits and prediction of different model

Sample	Model	Candidates				Prediction	Golden Answer
		A	B	C	D		
1	**MMN**	0.02	0.01	0.94	0.03	C	C
	EAMs	0.01	0.14	0.03	0.82	D	
	QPAM	0.29	0.06	0.53	0.12	C	
2	**MMN**	0.03	0.02	0.05	0.90	D	D
	EAMs	0.29	0.12	0.05	0.54	D	
	QPAM	0.01	0.10	0.78	0.11	C	

Related Work

Machine Reading Comprehension (MRC) has been studied extensively in the literature. The emergence of many large-scale datasets promotes the research in this field. (Hermann et al. 2015) generated a large cloze-style dataset from CNN news corpus automatically. (Rajpurkar et al. 2016) released Stanford Question Answer Dataset(SQuAD), where the question is generated by the human according to wikipedia articles and the answer is a span of passage. Compared to these datasets where answer is extracted from passage directly, answers for questions in MS-MARCO (Nguyen et al. 2016), RACE (Lai et al. 2017), DuReader (He et al. 2018) are human-generated, which is more challenging. This can be seen that current state-of-the-art model can only achieve almost 50% accuracy on RACE, though there are only 4 candidate answers for each question.

Most recent works involving multiple choice reading comprehension attempt to synthesize evidence representation according to passage and question by utilizing attention mechanism. (Parikh et al. 2018) proposes ElimiNet which tries to use soft-eliminating to exclude the incorrect candidate answers. However, it is usually ignored to model semantic relationship between evidence and answer deeply, which is expecially important when answer is a long sequence and almost complete sentence. Among existing works, DFN (Xu et al. 2017) and Hier-Co-Matching (Wang et al. 2018) are a bit similar to **MMN**. The key differences between **MMN** and them are three-fold: (1) **MMN** matches attended context sequences directly instead of using multi-perspective matching (Wang, Hamza, and Florian 2017), where the input feed to match is similarity score. (2) **MMN** explores not only triple sequences matching but pariwise sequences matching. This two matching are unified under our proposed compose-match framework. Furthermore, **MMN** aggregates the information with both max pooling and attentive pooling. (3) **MMN** employs sequential variant of highway network (Srivastava, Greff, and Schmidhuber 2015) to improve the performance further.

Conclusion

In this work, we propose a novel **M**ulti-**M**atching **N**etwork for multiple choice machine reading comprehension. Our **MMN** learns the relationship among passage, question and candidate answer from different ways. Both ways guide us to design our model which is intuitive and effective. Empirical results on RACE dataset demonstrate **MMN**'s effectiveness, which achieves state-of-the-art result. However, on RACE dataset, there is a huge significant performace gap compared to human performance. This indicates difficulty of task and huge improvement space of our model. In the future, we will try to incorporate commonsense knowledge to improve the model further.

References

Chen, D.; Bolton, J.; and Manning, C. D. 2016. A thorough examination of the cnn/daily mail reading comprehension task. In *ACL*, 2358–2367.

Chung, J.; aglar Gülehre; Cho, K.; and Bengio, Y. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *CoRR* abs/1412.3555.

Dhingra, B.; Liu, H.; Yang, Z.; Cohen, W. W.; and Salakhutdinov, R. 2017. Gated-attention readers for text comprehension. In *ACL*.

He, W.; Liu, K.; Liu, J.; Lyu, Y.; Zhao, S.; Xiao, X.; Liu, Y.; Wang, Y.; Wu, H.; She, Q.; Liu, X.; Wu, T.; and Wang, H. 2018. Dureader: a chinese machine reading comprehension dataset from real-world applications. In *ACL Workshop on Machine Reading for Question Answering*, 37–46.

Hermann, K. M.; Kociský, T.; Grefenstette, E.; Espeholt, L.; Kay, W.; Suleyman, M.; and Blunsom, P. 2015. Teaching machines to read and comprehend. In *NIPS*.

Kingma, D. P., and Ba, J. 2014. Adam: A method for stochastic optimization. *CoRR* abs/1412.6980.

Lai, G.; Xie, Q.; Liu, H.; Yang, Y.; and Hovy, E. 2017. Race: Large-scale reading comprehension dataset from examinations. In *EMNLP*.

Nguyen, T.; Rosenberg, M.; Song, X.; Gao, J.; Tiwary, S.; Majumder, R.; and Deng, L. 2016. MS MARCO: A human generated machine reading comprehension dataset. In *NIPS Workshop on Cognitive Computation*.

Parikh, S.; Sai, A.; Nema, P.; and Khapra, M. 2018. Eliminet: A model for eliminating options for reading comprehension with multiple choice questions. In *IJCAI*.

Rajpurkar, P.; Zhang, J.; Lopyrev, K.; and Liang, P. 2016. Squad: 100, 000+ questions for machine comprehension of text. In *EMNLP*.

Seo, M. J.; Kembhavi, A.; Farhadi, A.; and Hajishirzi, H. 2016. Bidirectional attention flow for machine comprehension. *CoRR* abs/1611.01603.

Srivastava, N.; Hinton, G. E.; Krizhevsky, A.; Sutskever, I.; and Salakhutdinov, R. 2014. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research* 15:1929–1958.

Srivastava, R. K.; Greff, K.; and Schmidhuber, J. 2015. Highway networks. *CoRR* abs/1505.00387.

Tay, Y.; Tuan, L. A.; and Hui, S. C. 2018. Multi-range reasoning for machine comprehension. *CoRR* abs/1803.09074.

Wang, S.; Yu, M.; Jiang, J.; and Chang, S. 2018. A co-matching model for multi-choice reading comprehension. In *ACL*, 746–751.

Wang, Z.; Hamza, W.; and Florian, R. 2017. Bilateral multi-perspective matching for natural language sentences. In *IJCAI*.

Xu, Y.; Liu, J.; Gao, J.; Shen, Y.; and Liu, X. 2017. Towards human-level machine reading comprehension: Reasoning and inference with multiple strategies. *CoRR* abs/1711.04964.

Yin, W.; Schütze, H.; Xiang, B.; and Zhou, B. 2016. ABCNN: attention-based convolutional neural network for modeling sentence pairs. *TACL* 4:259–272.

Zhu, H.; Wei, F.; Qin, B.; and Liu, T. 2018. Hierarchical attention flow for multiple-choice reading comprehension. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, New Orleans, Louisiana, USA, February 2-7, 2018*.