

Reference Knowledgeable Network for Machine Reading Comprehension

Yilin Zhao^{1,2,3}, Zhuosheng Zhang^{1,2,3}, Hai Zhao^{1,2,3}

¹Department of Computer Science and Engineering, Shanghai Jiao Tong University

²Key Laboratory of Shanghai Education Commission for Intelligent Interaction and Cognitive Engineering, Shanghai Jiao Tong University, Shanghai, China

³MoE Key Lab of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University, Shanghai, China
zhaoyilin@sjtu.edu.cn, zhangzs@sjtu.edu.cn, zhaohai@cs.sjtu.edu.cn

Abstract

Multi-choice Machine Reading Comprehension (MRC) is a major and challenging form of MRC tasks that requires model to select the most appropriate answer from a set of candidates given passage and question. Most of the existing researches focus on the modeling of the task datasets without explicitly referring to external fine-grained common-sense sources, which is a well-known challenge in multi-choice tasks. Thus we propose a novel reference-based knowledge enhancement model based on span extraction called *Reference Knowledgeable Network (RekNet)*, which simulates human reading strategy to refine critical information from the passage and quote external knowledge in necessity. In detail, *RekNet* refines fine-grained critical information and defines it as *Reference Span*, then quotes external knowledge quadruples by the co-occurrence information of *Reference Span* and answer options. Our proposed method is evaluated on two multi-choice MRC benchmarks: RACE and DREAM, which shows remarkable performance improvement with observable statistical significance level over strong baselines.

1 Introduction

Machine reading comprehension (MRC) is a challenging natural language understanding task which lets the machine predict appropriate answer to the question according to a given passage or document (Zhang et al., 2020c; Wang et al., 2019; Huang et al., 2019). According to answer styles, MRC tasks can be roughly divided into generative (to generate answer texts to given questions), extractive (to extract spans from given contexts to answer questions) and multi-choice (to select the most appropriate answer among given answer options) tasks (Baradaran et al., 2020). The multi-choice task is the focus of this work.

Recently, various datasets and tasks have been proposed, promoting a rapid improvement of MRC techniques (Lowe et al., 2015; Wang et al., 2018a; Rajpurkar et al., 2016). Early MRC datasets usually provide passages whose contents are extracted from articles (Rajpurkar et al., 2018; Lai et al., 2017; Huang et al., 2019). Recently, conversational reading comprehension has aroused great interests whose passages are derived from multi-turn dialogue segments (Reddy et al., 2019; Choi et al., 2018; Sun et al., 2019a; Zhang et al., 2018), making the task be more challenging.

The popular practice to solve MRC problems is adopting pre-trained language models (LM) as encoder module (Peters et al., 2018; Devlin et al., 2019; Clark et al., 2019; Lan et al., 2019). Instead of better exploiting pre-trained LMs, this paper is motivated by human reading strategies to decouple MRC into *sketchy reading* by extracting the critical spans from the passage, and *extensive reading* by seeking external knowledge. As a result, we propose a knowledge enhancement model based on extracted critical information called *RekNet (Reference Knowledgeable Network)*. In detail, the proposed *RekNet* refines the fine-grained critical information by a span extraction model and defines it as *Reference Span*, then quotes relevant external knowledge in the form of quadruples by the co-occurrence information of *Reference Span* and answer options. An example process of *RekNet* is shown in Figure 1.

In summary, our main contributions are follows:

- 1) We propose a novel reference-based knowledge enhancement model *RekNet*, which makes the first attempt to obtain fine-grained evidence for inference and knowledge retrieving on MRC tasks.
- 2) *RekNet* uses novel knowledge quadruples to quote relevant and credible knowledge.
- 3) *RekNet* is applied to two multi-choice MRC benchmarks, RACE (Lai et al., 2017) and DREAM

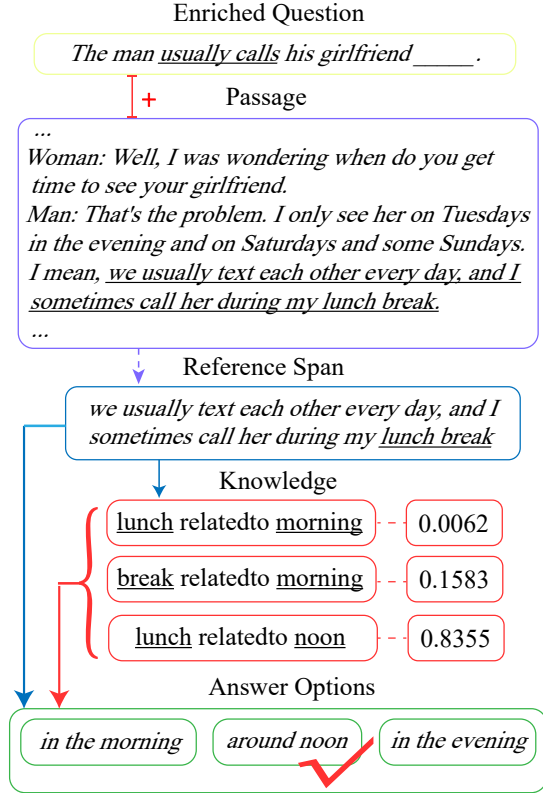


Figure 1: An example process of our model. After integrating the passage and enriched question, we refine *Reference Span* and obtain knowledge quadruples related to it, and select the appropriate answer. The weights for each knowledge quadruple will be explained in Section 6.2. In this example, the enriched question is the same as the original question.

(Sun et al., 2019a) and improves the performance of baseline models by 1.0% and 1.1% respectively, which both pass the significance test of MRC tasks.

2 Related Studies

For multi-choice MRC tasks, existing studies mostly focused on pre-training a powerful language model (Devlin et al., 2019; Liu et al., 2019b; Yang et al., 2019; Lan et al., 2019; Clark et al., 2019) as encoder, or improving the matching interactions between the input sequences based on attention methods (Zhang et al., 2020a; Zhu et al., 2020; Wang et al., 2018b; Tang et al., 2019; Zhang et al., 2020b).

Recently, some researchers attempted to use other methods to improve the performance of MRC tasks other than exploiting the pre-trained LM, among which *knowledge injection* and *reading strategy simulation* are one of the most popular ones.

Knowledge Injection

Observing the drawbacks of the lack of common-sense in MRC models, some researchers attempted to inject external knowledge (Bhagavatula et al., 2020; Lin et al., 2020a; Schwartz et al., 2020). Xia et al. (2019) used auxiliary tasks to obtain relevant knowledge, while Mihaylov and Frank (2018) encoded external commonsense knowledge as key-value memory in a cloze-style setting to inject extra knowledge into their model. Lin et al. (2019) quoted a conceptual subgraph of external knowledge to get better performance. For non-MRC dialogue task, Chaudhuri et al. (2018) used GRUs to refine and encode keywords from passages, then combined domain-specific knowledge. Lin et al. (2020b) proposed a constrained text generation task for generative commonsense reasoning.

With the impact of knowledge noise on model performance became more obvious, many researchers also proposed their methods to quote relevant knowledge and reduce knowledge noise (Kim et al., 2020; Liu et al., 2019a).

However, though all these studies attempted to select relevant knowledge, they generally ignored the powerful utility of critical information to knowledge injection as well as the filtration of untrustworthy knowledge.

Reading Strategy Simulation

Inspired by human reading strategies, some researchers tried to teach their models to follow specific reading strategies to improve MRC performance (Li et al., 2018). Zhang et al. (2021) proposed a retrospective reader for span-based MRC tasks, and Sun et al. (2019b) applied three specific human reading strategies.

Among all reading strategies, *evidence information extraction* has been paid great attention (Choi et al., 2017). Wang et al. (2019) gave the first attempt to extract evidence sentences in a multi-choice MRC task, and Yadav et al. (2019) applied this method to multi-hop QA problem. Niu et al. (2020) supervised the evidence extractor with auto-generated evidence labels in an iterative process. All these studies perform information exaction only on a sentence-level granularity.

Our Method

This work differs from previous studies by two main aspects:

1) To highlight question-aware critical information from context, our method models the *Reference*

Span based on span extraction, instead of using the whole context for inference or external knowledge retrieving;

2) To alleviate the negative impact of untrustworthy knowledge, our method quotes relevant external knowledge in the form of quadruples, instead of knowledge triples used in previous studies.

To our best knowledge, our model is the first *reference-based knowledge enhancement* model in multi-choice MRC tasks.

3 Preliminary Experiments

To learn general characteristics of multi-choice MRC tasks, we randomly extracted 50 examples in DREAM and RACE respectively, finding that 38% examples of DREAM and 28% of RACE can be inferred just by several *adjacent phrases* in one single sentence. This finding indicates that concise span such as phrases often contains the salient information which leads to correct answers for multi-choice MRC tasks.

Model	Dev	Test
Baseline	65.74	65.56
+ Reference	67.65	67.86
Reference only	59.02	58.94

Table 1: Preliminary experiments for DREAM task. The results are based on ALBERT_{base}. The *Baseline* is fed with the triple of $\{passage, question, answer options\}$; *Reference* denotes the *Reference Span* extracted by a span-based MRC model trained on SQuAD v2.0. *+ Reference* represents that there are two sequence triples $\{passage, question, answer options\}$ and $\{reference, question, answer options\}$. Their pooled logits are concatenated for prediction to both take advantage of the original redundant and refined critical information. *Reference only* means the input is $\{reference, question, answer options\}$.

Inspired by this finding, we directly used the extracted *Reference Spans* to replace the passages as input, and the model achieved good enough performance as Table 1 shows. From the results of this preliminary experiment, there are two main reasons which may cause unsatisfactory performance according to our observation: 1) harder questions that require further reasoning (Figure 1); 2) potential mistakes from the span extraction. The first issue could be handled by augmentation from extra sources¹ and the second issue could be alleviated

¹We found that 26% examples of DREAM and 20% of RACE needing external knowledge to answer questions.

by both modeling the original passage-aware and the new reference-aware sequences as the superior result (+2.30%) of *+ Reference* shown in Table 1.

To further verify whether fine-grained level information has a positive impact to information extraction, we designed more experiments in Appendix A.1. The improvement comparing to coarse-grained information baseline proves the superiority of *Reference Span*.

Inspired by both our findings above and human reading and understanding experience (Ding et al., 2019; Zhang et al., 2021), we design our model following a two-stage reading strategy to deal with multi-choice MRC tasks. As the human reading pattern, one will read the passage and highlight the critical information for the given question in a sketchy reading first, then integrate extensive relevant knowledge sources to extrapolate the appropriate answer. In detail, our reading strategy consists of two steps: critical information extraction and knowledge injection. The process is implemented as follows:

i) In the task perspective, the machine should focus on question-related information from the lengthy passage, which can interpret the process of human reading comprehension. We call it *sketchy reading*, which embodies *reading* process in reading comprehension.

ii) In the model perspective, the machine should solve the given questions with transcendental external knowledge and the current context. We call it *extensive reading*, which embodies *comprehension* process in MRC.

In *extensive reading*, due to MRC tasks are rich in text content and model may quote a large amount of irrelevant or untrustworthy knowledge, we quote knowledge in the form of quadruples to filter distracting information².

Inspired by above findings and analysis, we stage effective module inside MRC model which conducts *critical information extraction* and *knowledge injection*.

4 Our Model

Multi-choice MRC tasks can be defined as a triplet (P, Q, A) , where P is passage, Q is question and A is a set of answer options: $A = A_1, \dots, A_n$, where n is the number of answer options for each question. Let $A_{correct} \in A$ be the correct answer of the given

²The comparison to triples used in previous studies can be found in Section 6.2.

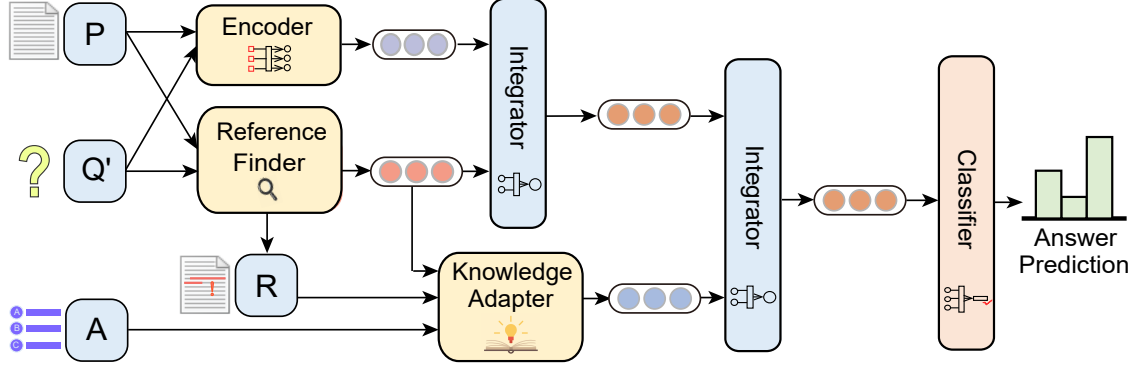


Figure 2: The overview of our model. P , Q' , A , R stand for *Passage*, *Enriched Question*, *Answer Options* and *Reference Span* respectively, Figure 3 and Figure 4 are the same. Our model reads one triplet (P, Q', A) for each time. Integrator No.1 simulates human *sketchy reading* while Integrator No.2 simulates human *extensive reading*.

question, our aim is to make $P(A_{correct} | P, Q, A)$ be the largest one in $P(A_i | P, Q, A), i \in (1, \dots, n)$ which represents the probability of each answer option.

Our model will refer to another element R , which is implemented as the critical information span (named as *Reference Span*) for each question, and we add co-occurrence information of all answer options A to question Q , getting enriched question Q' for our model. One highlight of our model is that we obtain a fine-grained reference source, which is the reason we call it *Reference Span* not *Reference Sentence*. Therefore our model defines multi-choice MRC tasks to a quadruple (P, Q', R, A) instead.

The overall structure of our model is showed in Figure 2. Our model consists of three modules: Reference Finder, Knowledge Adapter and Integrator. We will introduce the details of these modules in the following subsections.

4.1 Reference Finder

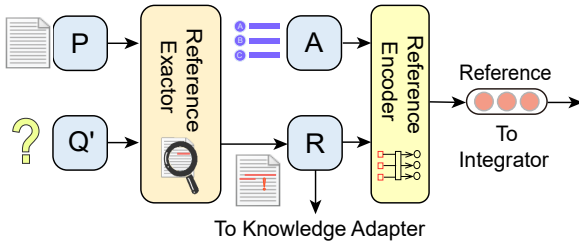


Figure 3: The overview of Reference Finder.

Reference Finder is to obtain *Reference Span* in sketchy reading like humans locate where they should focus on in a lengthy passage. In this work we only refine one *Reference Span* for RACE and DREAM, because these MRC tasks do not need

multi-hop inference like MultitRC (Khashabi et al., 2018) or HotpotQA (Yang et al., 2018) and a large proportion of questions can be answered by several adjacent phrases as we show in section Preliminary Experiments.

In detail, Reference Finder takes the initial passage P , enriched question Q' , and answer options A as input, produces *Reference Span* and encodes it for Knowledge Adapter and Integrator. In this module, we have two main components called Reference Extractor and Reference Encoder, as Figure 3 shows.

Enriched Question Q'

One of the differences between multi-choice and other MRC tasks is that there is also some critical information in given answer options. This may cause that the model cannot obtain precise *Reference Span* without referring answer options. So for each question, we pick up the co-occurrence information of all answer options and add them to the tail of the original question Q in order, forming the enriched question Q' . This action facilitates Reference Finder to obtain *Reference Span* more precisely. More details and studies on *Enriched Question* is shown in Appendix A.2.

Reference Extractor

Taking concatenation of the Enriched Question Q' and original passage P as input, Reference Extractor exploits their representations to select *Reference Span*. We define the hidden size of pre-trained language model as H and the final hidden vector for the i -th input token as $T_i \in \mathbb{R}^H$, then we score the *Reference Span* starts from i -th input token and ends at j -th input token as $S \cdot T_i + E \cdot T_j$, where S is the start vector while E is the end vector, S

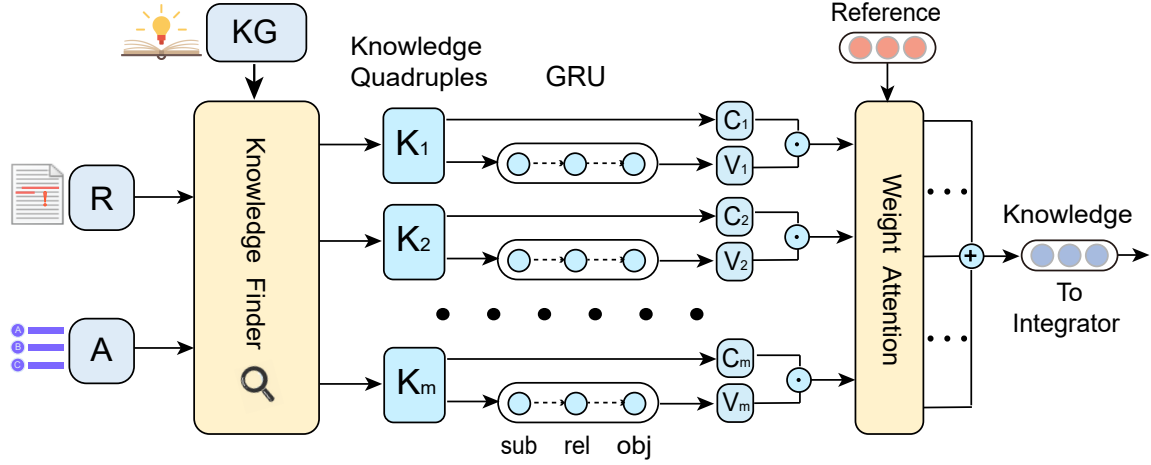


Figure 4: The overview of Knowledge Adapter. KG stands for *Knowledge Graph*. m is the number of knowledge quadruples, which is equal to $k \times n$ in our model. K_i, C_i and V_i refer to the i -th knowledge quadruple, the i -th confidence value and the last hidden state vector of the i -th input embedding. \cdot means scalar multiplication operation.

and $E \in \mathbb{R}^H$, like (Devlin et al., 2019). The largest i and j for the former formula are chosen as the start and end token ids, and the contents between i -th input token and j -th input token are chosen as *Reference Span* in original order. In case of Reference Extractor returning null *Reference Span*, which may impact the input of Knowledge Adapter, we simply discard the returned null span.

Reference Encoder

Reference Encoder is a pre-trained LM (we use ALBERT_{base} for it) and encodes each answer option A_i with *Reference Span* R in the form of $[CLS] R [SEP] A_i [SEP]$, making RekNet gain information from *Reference Span* directly. With Reference Encoder, we get a set of Reference Vectors $RV = RV_1, \dots, RV_n$ and $RV \in \mathbb{R}^{H \times n}$. At least, the resulted Reference Vector will fuse with Passage Vector PV which is produced by *Encoder* (top-left of Figure 2).

4.2 Knowledge Adapter

Knowledge Adapter quotes external knowledge for *RekNet*. It takes *Reference Span* R , answer options A and an external knowledge graph as input, produces encoded knowledge quadruples. The structure of Knowledge Adapter is shown in Figure 4, which can be divided into two modules: Knowledge Finder and Knowledge Encoder.

Knowledge Finder

Knowledge Finder finds knowledge quadruples for each P-Q-A triplet, and due to passage usually has a lengthy context that contains a lot of noise, we send

R-A binaries as input instead. In detail, Knowledge Finder quotes knowledge facts whose *subject entity* and *object entity* exist in *Reference Span* and *Answer Option* respectively, to remove irrelevant knowledge for the given question. We do a prototype matching with them³. Knowledge facts are saved in the form of quadruples (sub, rel, obj, con), which stands for the subject, relation, object and confidence value of each knowledge fact. (*doctor*; *capableof*; *help_sick_person*, 4.472) is an example quadruple, where the subject, relation, and object can be a word or a phrase, and confidence value is a number larger than 0.1⁴. Larger confidence value indicates the knowledge is more trustworthy.

We set each R-A binary can only obtain the number of $k \times n$ knowledge quadruples with the largest confidence values, where k is the number of quotable knowledge quadruples for each answer option, and this method can reduce knowledge noise significantly. If one R-A binary can not get enough quadruples, we add null quadruples whose confidence values are 0 to fill up to a specified amount.

Knowledge Encoder

Knowledge Encoder is the remaining part of Knowledge Adapter. To encode all knowledge quadruples for one R-A binary, we firstly get input embedding for each word or phrase appears in knowledge quadruples, then we use a GRU to

³We also do a cosine similarity matching, but there is not an significant improvement compared to this simple method.

⁴In this way *RekNet* can significantly separate knowledge quadruples with low confidence values from blank knowledge quadruples.

encode the quadruple as following:

$$hidden_{sub} = GRU(emb_{sub}, 0) \quad (1)$$

$$hidden_{rel} = GRU(emb_{rel}, hidden_{sub}) \quad (2)$$

$$hidden_{obj} = GRU(emb_{obj}, hidden_{rel}) \quad (3)$$

where emb_{sub} , emb_{rel} , emb_{obj} are the input embeddings of subject, relation and object and $hidden_{sub}$, $hidden_{rel}$, $hidden_{obj}$ are the last hidden states of GRU. The motivation of this encoding method is that we can retain the directionality of input quadruples, and encode external knowledge in the same vector space as the plain tokens.

After we get the embeddings of the quadruples, we send all the $k \times n$ embeddings to a weighted attention module with their confidence values, and the main operation can be expressed as the following formula:

$$KV = \sum_{i=1}^{k \times n} Softmax(WeightAtt(RV, h_i, c_i))^T h_i \quad (4)$$

where h_i and c_i are the $hidden_{obj}$ and confidence value of the i -th knowledge quadruple for a given question, KV and RV is the Knowledge Vector and Reference Vector of each given question, which is same for all answer options of the question. $WeightAtt$ is an attention function to compute the final score (or attention weight) for each knowledge quadruple. In detail, we do a matrix multiplication for RV and h_i , then do a scalar multiplication for the result and c_i .

Through the operations of Knowledge Adapter, we obtain the encoded knowledge quadruples for each R-A binary and send them to Integrator.

4.3 Integrator

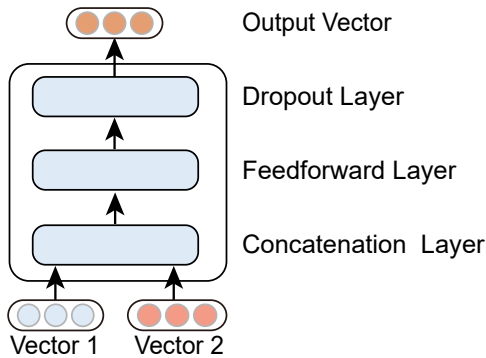


Figure 5: The overview of Integrator.

There are two integrators in our model whose aim are to integrate two embedding vectors to a

fusion embedding vector, and all the vectors are in the same vector space. The structure of Integrator is illustrated in Figure 5.

In detail, there are three layers in Integrator. In Concatenation Layer, two input embedding vectors in size of H are spliced into a vector in size of $2 \times H$. Then Integrator reduces the dimension of spliced vector to hidden size H in Feedforward Layer, which is a linear layer. Ultimately, there is a Dropout Layer to prevent overfitting.

5 Experiments

5.1 Setup

The implementation of our model was based on the Pytorch implementation of ALBERT_{xxlarge}. In the experiment, we adopted ALBERT (Lan et al., 2019), which was trained on the SQuAD v2.0 (Rajpurkar et al., 2018) as the *Reference Span* extraction model for Reference Finder⁵, and set the number of knowledge quadruples for each answer option k to 5, which leads to the best performance for RekNet⁶. The finetuning hyperparameters of RekNet are shown in Appendix A.3.

5.2 Dataset

Task Datasets

We used RACE (Lai et al., 2017) and DREAM (Sun et al., 2019a) as our evaluation datasets. Both of the datasets are collected from English examinations.

RACE is a large-scale MRC task. Each passage has multiple questions, and each question has four answer options. Most questions need contextual reasoning, and the domains of passages are diversified.

DREAM is a dialogue-based dataset for multi-choice MRC. Each dialogue has multiple questions, and each question has three answer options. The challenge of the dataset is that more than 80% of the questions are non-extractive and require reasoning from multi-turn dialogues, and more than a third of given questions involve commonsense knowledge.

⁵During extractive tasks, SQuAD v2.0 has the largest contribution to other MRC tasks, claimed by Khashabi et al. (2020).

⁶During experiments we found that smaller k could lead to the loss of potentially important knowledge, while larger k might bring knowledge noise and unnecessary computational consumption. When $k = 4$ and 6, the performance on ALBERT_{base} drops by 0.70% and 0.44% on DREAM compared to $k = 5$.

Knowledge Source To quote commonsense knowledge to our model, we utilized ConceptNet 5.7.0 (Speer et al., 2017), one of the most largest structured knowledge base with the confidence value for each knowledge fact as the knowledge source. Due to the huge size of ConceptNet, we only remained its English concepts and merged some relations which share similar meanings. To obtain the input embedding for each word or phrase, we used the pre-trained 100 dimension of GloVe (Pennington et al., 2014) embedding vectors.

5.3 Results

Model	Dev	Test
FTLM++	58.1	58.2
BERT _{large}	66.0	66.8
XLNet	-	72.0
RoBERTa _{large}	85.4	85.0
RoBERTa _{large} + MMM	88.0	88.9
ALBERT _{xxlarge} + DUMA	89.3	90.4
ALBERT _{xxlarge} + DUMA + MTL	-	91.8
Baseline (ALBERT _{xxlarge})	89.2	88.5
RekNet	89.8	89.6

Table 2: The results on DREAM dataset. All the results except our implementations (last row) are from the leaderboard. MTL denotes multi-task learning.

Model	Test (M/H)
XLNet	81.8 (85.5/80.2)
XLNet + DCMN+	82.8 (86.5/81.3)
RoBERTa _{large}	83.2 (86.5/81.8)
DCMN+ (ensemble)	84.1 (88.5/82.3)
RoBERTa _{large} + MMM	85.0 (89.1/83.3)
ALBERT _{xxlarge} + DUMA	88.0 (90.9/86.7)
Megatron-BERT _{3.9B}	89.5 (91.8/88.6)
Baseline (ALBERT _{xxlarge})	86.5 (89.0/85.5)
RekNet	87.5 (90.1/86.2)

Table 3: Results on RACE dataset. All the results except our implementations (last row) are from the leaderboard. The score on dev set is 87.8%.

We used accuracy as evaluation criteria for multi-choice MRC tasks. Tables 2-3 show the results of our model compared with the baselines, as well as other public models on the DREAM and RACE tasks respectively. As a supplement, we got 3.1% and 3.2% performance improvement on RACE and DREAM compared to baseline respectively, when we implemented our model on ALBERT_{base}. Sig-

nificant test shows that our model is significantly better than the baseline with p-value < 0.01. In addition, we only introduce approximately 33M additional parameters compared with the 244M in the baseline, which demonstrates the conciseness and efficiency of our model.

6 Analysis

We used our model on ALBERT_{base} for the following analysis. All the analyses are based on DREAM task. Analysis of Error Cases are shown in Appendix A.4 due to the limited space.

6.1 Ablation Studies

Model	Dev	Test
Baseline (ALBERT _{base})	65.74	65.56
RekNet	68.04	68.74
- RF	66.72	67.76
- KA	67.65	67.86
Relocation Model	67.94	68.45
Masked Model ($\beta = 0$)	68.28	68.05

Table 4: The results of ablation and degeneration studies for DREAM task. RF: Reference Finder. KA: Knowledge Adapter.

As Figure 2 shows, we added two modules (Reference Finder and Knowledge Adapter) to *RekNet*. To study the importance of each module, we removed one of them for each time, keeping the parameters unchanged, obtaining results in Table 4. It indicates that removing Reference Finder brings 0.98% performance loss to the intact model while Knowledge Adapter brings 0.88%, which shows both modules are indispensable to our *RekNet*. This finding further demonstrates the rationality of the reading strategy *RekNet* follows:

i) After removing Reference Finder, Knowledge Adapter can only blindly perform the extensive reading in full-text range, quoting a large amount irrelevant knowledge and knowledge noise to *RekNet*, making it suffer from lacking *reading* process.

ii) After removing Knowledge Adapter, *RekNet* cannot introduce external knowledge separated from the text, leading to the lack of relevant necessary information and cannot reflect the *comprehension* process.

To study the impact of the fusion order, we exchanged the order of Reference Vector and Knowledge Vector integrates with Passage Vector, which could be considered to exchange the position of two

integrators. As Table 4 shows, relocation of these two modules may bring bad effect to our model, which may because:

i) The context of *Reference Span* is more similar to the initial passage context, and knowledge quadruples are obtained from *Reference Span*. Integrating two embeddings that represent a more similar context can learn more features than combining two more different embeddings.

ii) Relocation Model violates the order of natural human reading comprehension strategy, which may make the Relocation Model quote some irrelevant knowledge facts and over-analyze them.

6.2 Analysis of Knowledge Quadruples

In the usage of knowledge, one of the highlights of our work is that we quote knowledge quadruples to our model rather than knowledge triples, which are widely used by other studies (Xia et al., 2019; Mihaylov and Frank, 2018; Lin et al., 2019; Chaudhuri et al., 2018). To prove the effectiveness of knowledge quadruples we proposed, we conducted a comparative experiment. We degenerated our knowledge quadruples to knowledge triples as baseline, by replacing the confidence values in knowledge quadruples with the values of 0-1 Mask Vectors.

In detail, Mask Vector is in the vector space of $\mathbb{R}^{k \times n}$ for each given R-A binary, and if we define Mask Vector as MV , MV_i is the i -th element of MV , we set:

$$MV_i = \begin{cases} 1 & c_i > \beta, \\ 0 & \text{Otherwise.} \end{cases} \quad (5)$$

for Mask Vector, where β is a non-negative threshold. We set $\beta = 0$ to degenerate knowledge quadruples into triples (treat all knowledge facts equally). The result is shown in Table 4. It indicates that Mask Vector performs worse than confidence values because there is a huge amount of untrustworthy knowledge facts in the knowledge graph, such as (*abdomen*, *relatedto*, *thorax*, 0.102). It is unreasonable to give them equal treatment to trustworthy ones.

We also do experiments over different thresholds are shown in Figure 6. It shows that during different thresholds, *Masked Model* performs well than *RekNet* without Knowledge Adapter (no knowledge), but worse than *RekNet* (use knowledge quadruples). The reasons for the trend of model performance changing with β may be that, with

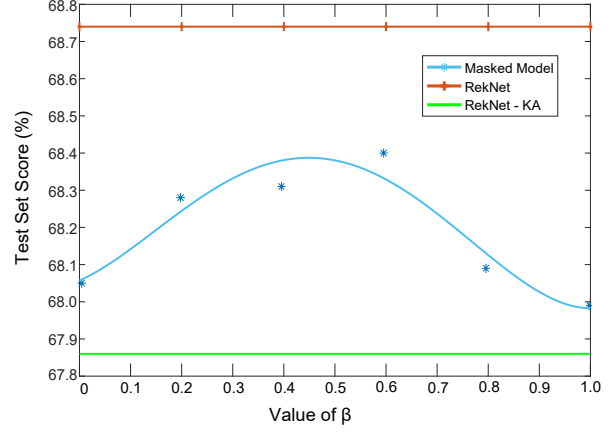


Figure 6: The test scores of *Masked Model* with different thresholds (β) on DREAM. KA: Knowledge Adapter.

low β , model cannot effectively filter out untrustworthy knowledge while with high β , model may filter out some credible and importance knowledge. In summary, all the results prove the effectiveness of confidence values.

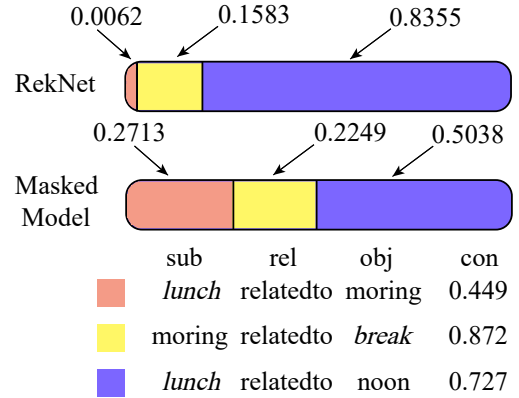


Figure 7: The final score (attention weight) of each knowledge fact for the dialogue in Figure 1. *RekNet* uses knowledge quadruples while *Masked Model* uses knowledge triples. In this example, red, yellow and blue knowledge fact is the untrustworthy, irrelevant and precise knowledge fact, respectively.

To show the effectiveness of knowledge filtering method of *RekNet* (using quadruples), we extracted the final score (attention weight) of each knowledge fact for one detailed example as Figure 7 shows, together with *Masked Model* (using mainstream triples).

As it shows, by using knowledge quadruples, *RekNet* gives the lowest score to (*lunch*, *relatedto*, *moring*, 0.449) because it is untrustworthy, and gives (*moring*, *relatedto*, *break*, 0.872) a low score in spite of its high confidence value, because it has

less contextual relevance to the *Reference Span*.

Compared with *RekNet*, model using knowledge triples will pay more attention to untrustworthy knowledge and less attention to precise ones, leading to the wrong prediction for question.

7 Conclusions

To alleviate the challenge of knowledge role missing in multi-choice MRC, this work makes the first attempt to integrating *external knowledge* based on *span extraction* into MRC modeling, presenting *Reference Knowledgeable Network (RekNet)*, which can simulate the human strategy of reading comprehension and quote external knowledge for multi-choice MRC tasks. *RekNet* helps achieve significantly performance improvement on two multi-choice MRC benchmarks RACE and DREAM, which passed the significance test. In the future, we will apply *RekNet* to other forms of MRC tasks.

References

- Razieh Baradaran, Razieh Ghiasi, and Hossein Amirkhani. 2020. A survey on machine reading comprehension systems. *arXiv preprint arXiv:2001.01582*.
- Chandra Bhagavatula, Ronan Le Bras, Chaitanya Malaviya, Keisuke Sakaguchi, Ari Holtzman, Hannah Rashkin, Doug Downey, Wen tau Yih, and Yejin Choi. 2020. Abductive commonsense reasoning. In *ICLR*.
- Debanjan Chaudhuri, Agustinus Kristiadi, Jens Lehmann, and Asja Fischer. 2018. Improving response selection in multi-turn dialogue systems by incorporating domain knowledge. In *CoNLL*, pages 497–507.
- Eunsol Choi, He He, Mohit Iyyer, Mark Yatskar, Wen-tau Yih, Yejin Choi, Percy Liang, and Luke Zettlemoyer. 2018. QuAC: Question answering in context. In *EMNLP*, pages 2174–2184.
- Eunsol Choi, Daniel Hewlett, Jakob Uszkoreit, Illia Polosukhin, Alexandre Lacoste, and Jonathan Berant. 2017. Coarse-to-fine question answering for long documents. In *ACL*, pages 209–220.
- Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. 2019. Electra: Pre-training text encoders as discriminators rather than generators. In *ICLR*.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, pages 4171–4186.
- Ming Ding, Chang Zhou, Qibin Chen, Hongxia Yang, and Jie Tang. 2019. Cognitive graph for multi-hop reading comprehension at scale. In *COLING*, pages 2694–2703.
- Lifu Huang, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2019. Cosmos qa: Machine reading comprehension with contextual commonsense reasoning. In *EMNLP*.
- Daniel Khashabi, Snigdha Chaturvedi, Michael Roth, Shyam Upadhyay, and Dan Roth. 2018. Looking beyond the surface: a challenge set for reading comprehension over multiple sentences. In *NAACL*.
- Daniel Khashabi, Sewon Min, Tushar Khot, Ashish Sabharwal, Oyvind Tafjord, Peter Clark, and Han-naneh Hajishirzi. 2020. UNIFIEDQA: Crossing format boundaries with a single QA system. In *Findings of EMNLP*.
- Byeongchang Kim, Jaewoo Ahn, and Gunhee Kim. 2020. Sequential latent knowledge selection for knowledge-grounded dialogue. *ICLR*.
- Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. 2017. RACE: Large-scale Reading comprehension dataset from examinations. In *EMNLP*, pages 785–794.
- Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. ALBERT: A lite BERT for self-supervised learning of language representations. *ICLR*.
- Weikang Li, Wei Li, and Yunfang Wu. 2018. A unified model for document-based question answering based on human-like reading strategy. *AAAI*.
- Bill Yuchen Lin, Xinyue Chen, Jamin Chen, and Xiang Ren. 2019. KagNet: Knowledge-aware graph networks for commonsense reasoning. In *EMNLP-IJCNLP*, pages 2829–2839.
- Bill Yuchen Lin, Haitian Sun, Bhuwan Dhingra, Manzil Zaheer, Xiang Ren, and William W. Cohen. 2020a. Differentiable open-ended commonsense reasoning. *arXiv preprint arXiv:2010.14439*.
- Bill Yuchen Lin, Wangchunshu Zhou, Ming Shen, Pei Zhou, Chandra Bhagavatula, Yejin Choi, and Xiang Ren. 2020b. CommonGen: A constrained text generation challenge for generative commonsense reasoning. In *Findings of EMNLP*, pages 1823–1840.
- Weijie Liu, Peng Zhou, Zhe Zhao, Zhiruo Wang, Qi Ju, Haotang Deng, and Ping Wang. 2019a. K-bert: Enabling language representation with knowledge graph. *arXiv preprint arXiv:1909.07606*.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019b. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

- Ryan Lowe, Nissan Pow, Iulian Serban, and Joelle Pineau. 2015. The Ubuntu dialogue corpus: A large dataset for research in unstructured multi-turn dialogue systems. In *SIGDIAL*, pages 285–294.
- Todor Mihaylov and Anette Frank. 2018. Knowledgeable reader: Enhancing cloze-style reading comprehension with external commonsense knowledge. In *COLING*, pages 821–832.
- Yilin Niu, Fangkai Jiao, Mantong Zhou, Ting Yao, Jingfang Xu, and Minlie Huang. 2020. A self-training method for machine reading comprehension with soft evidence extraction. *ACL*.
- Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*, pages 1532–1543.
- Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *COLING*.
- Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don’t know: Unanswerable questions for SQuAD. In *ACL*, pages 784–789.
- Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *EMNLP*, pages 2383–2392.
- Siva Reddy, Danqi Chen, and Christopher D Manning. 2019. CoQA: A conversational question answering challenge. *TACL*, 7:249–266.
- Vered Shwartz, Peter West, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2020. Unsupervised commonsense question answering with self-talk. In *EMNLP*, pages 4615–4629.
- Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. *AAAI*, 31:4444–4451.
- Kai Sun, Dian Yu, Jianshu Chen, Dong Yu, Yejin Choi, and Claire Cardie. 2019a. DREAM: A challenge data set and models for dialogue-based reading comprehension. *TACL*, 7:217–231.
- Kai Sun, Dian Yu, Dong Yu, and Claire Cardie. 2019b. Improving machine reading comprehension with general reading strategies. In *NAACL*, pages 2633–2643.
- Min Tang, Jiaran Cai, and Hankz Hankui Zhuo. 2019. Multi-matching network for multiple choice reading comprehension. In *AAAI*, volume 33, pages 7088–7095.
- Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018a. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *EMNLP Workshop*, pages 353–355.
- Hai Wang, Dian Yu, Kai Sun, Jianshu Chen, Dong Yu, David McAllester, and Dan Roth. 2019. Evidence sentence extraction for machine reading comprehension. *SIGLL*.
- Shuohang Wang, Mo Yu, Jing Jiang, and Shiyu Chang. 2018b. A co-matching model for multi-choice reading comprehension. In *COLING*, pages 746–751.
- Jiangnan Xia, Chen Wu, and Ming Yan. 2019. Incorporating relation knowledge into commonsense reading comprehension with multi-task learning. *CIKM*.
- Vikas Yadav, Steven Bethard, and Mihai Surdeanu. 2019. Quick and (not so) dirty: Unsupervised selection of justification sentences for multi-hop question answering. In *EMNLP-IJCNLP*, pages 2578–2589.
- Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *NIPS*, pages 5754–5764.
- Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *EMNLP*.
- Shuailiang Zhang, Hai Zhao, Yuwei Wu, Zhuosheng Zhang, Xi Zhou, and Xiang Zhou. 2020a. DCMN+: Dual co-matching network for multi-choice reading comprehension. In *AAAI*.
- Zhuosheng Zhang, Jiangtong Li, Pengfei Zhu, and Hai Zhao. 2018. Modeling multi-turn conversation with deep utterance aggregation. In *COLING 2018*, pages 3740–3752.
- Zhuosheng Zhang, Yuwei Wu, Hai Zhao, Zuchao Li, Shuailiang Zhang, Xi Zhou, and Xiang Zhou. 2020b. Semantics-aware bert for language understanding. In *AAAI*.
- Zhuosheng Zhang, Junjie Yang, and Hai Zhao. 2021. Retrospective reader for machine reading comprehension. In *AAAI*.
- Zhuosheng Zhang, Hai Zhao, and Rui Wang. 2020c. Machine reading comprehension: The role of contextualized language models and beyond. *arXiv preprint arXiv:2005.06249*.
- Pengfei Zhu, Hai Zhao, and Xiaoguang Li. 2020. Dual multi-head co-attention for multi-choice reading comprehension. *arXiv preprint arXiv:2001.09415*.

A Appendix

A.1 Fine-grained Comparison Experiment

Method	Dev	Test
TF-IDF Method	56.60	55.44
Our Method (sentence)	58.82	58.37
Our Method	59.02	58.94

Table 5: The results of fine-grained information extraction experiments for DREAM task. The results are based on ALBERT_{base}. The settings of *Our Method* are same as *Reference only* in Table 1, and *Our Method (sentence)* expands the fine-grained information (Reference Span) to the entire sentence. As to *TF-IDF*, we chose sentences whose scores are larger than $0.7 \times$ the largest appearing score as the input *reference*, while other inputs are same as *Our Method*.

We designed two baselines to obtain reference sentences on DREAM, one calculated TF-IDF scores of each sentence in passage and given question, while another one used our LM encoder to get whole reference sentences. The results as Table 5 shows. With *Reference Span* we extracted, the baseline got extra 0.57% improvement comparing to general coarse-grained information extraction method.

A.2 Details and Studies on Enriched Question Q'

To show the advantages of *Q'* compared to *Q*, we selected one detailed example in DREAM. As Dialogue 1 on Table 6 shows, only we know the key point to the question is *working situation of the man*, we can get *Reference Span* accurately.

The situation in Dialogue 1 is not rare. Take training set of DREAM for example, there are 61.1% questions getting enriched to obtain *Reference Span* s more accurately as Table 6 shows, and the average length of added tokens is 1.84. It shows for most questions, original questions will be slightly enriched by co-occurrence information, which may provide critical words without too much noise due to the length of added tokens.

And to learn the improvement brought by enriched information, we did a degradation experiment as Table 7 shows. The results show that *Q'* can improve the performance of MRC model and the main contribution of *Q'* is to help model get more precise *Reference Spans*. And to expand *RekNet* to other types of MRC tasks such as extractive MRC task, we can degenerate *Q'* into *Q*.

Dialogue 1
...
W: <i>You worked for a large company before, didn't you?</i>
M: <i>Yes, I did. But I prefer a small company.</i>
W: <i>Is it really different?</i>
M: <i>Oh, yes. It's much different. I like a small company because it's more exciting.</i>
...
Q: <i>What do we learn from the conversation?</i>
A. <i>The man has been working in a small company for a long time.</i>
B. <i>The man used to work for a big company, but now he works in a small one.</i> (correct answer)
C. <i>The man works in a small company, but he doesn't like it.</i>
Q': <i>What do we learn from the conversation?</i>
<i>The man works in a small company.</i>

Table 6: Sample dialogue of DREAM dataset. It shows the situation that only model knows what the key point is to the question from answer options, it can refine the critical information span accurately.

Q or Q'	Dev	Test
Q' to All Modules	68.04	68.74
Q to All Modules	67.60	68.12
Q' to RF and Q to Other Modules	68.46	68.50

Table 7: The results of question degradation experiments for DREAM task. The results are based on *RekNet* on ALBERT_{base}. RF: *Reference Finder* module.

A.3 Finetuning Hyperparameters

Our finetuning hyperparameters for RACE and DREAM are given in Table 8, which leads to the best performance.

A.4 Error Case Analysis

We extracted 50 error cases of *RekNet* based on ALBERT_{base} on DREAM randomly, finding 36% are related to logic calculation (especially numerical calculation)⁷, as Table 9 shows. Though *Reference Span* in *RekNet* can extract the computing words like *discount*, *half* from passages and questions, it fails to calculate the correct result due to lack of human numeral logic operation, which calls for more in-depth researches in MRC field.

⁷Original DREAM and RACE dataset have 14% and 8% examples related to logic calculation.

Hyperparam	RACE	DREAM
Learning Rate	1e-5	1e-5
Batch Size	32	24
Warmup Steps	1000	50
Maximum Sentence Length	512	512
Maximum <i>Reference Span</i> Length	512	256
Training Epochs	2	2
Steps to Save Checkpoints	3000	382

Table 8: The finetuning hyperparameters of *RekNet*.

Dialogue 2
<p>...</p> <p><i>W: The price for one person for a ten-day tour is only \$1,088, which includes round-trip airfare.</i></p> <p><i>M: That sounds reasonable. By the way, do you have a discount for two?</i></p> <p><i>W: Yes, you can have a 10% discount.</i></p> <p><i>Q: If the man and his wife go on the recommended package tour, how much should they pay?</i></p> <p>A. \$1,088.</p> <p>B. \$1,958. (correct answer)</p> <p>C. \$2,176.</p>

Table 9: Sample dialogue related to numeral calculation of DREAM dataset.