Reference Knowledgeable Network for Machine Reading Comprehension

Yilin Zhao, Zhuosheng Zhang, Hai Zhao

Abstract—Multi-choice Machine Reading Comprehension (MRC) as a challenge requires models to select the most appropriate answer from a set of candidates with a given passage and question. Most of the existing researches focus on the modeling of specific tasks or complex networks, without explicitly referring to relevant and credible external knowledge sources, which are supposed to greatly make up for the deficiency of the given passage. Thus we propose a novel reference-based knowledge enhancement model called Reference Knowledgeable Network (RekNet), which simulates human reading strategies to refine critical information from the passage and quote explicit knowledge in necessity. In detail, RekNet refines fine-grained critical information and defines it as Reference Span, then quotes explicit knowledge quadruples by the co-occurrence information of Reference Span and candidates. The proposed RekNet is evaluated on three multi-choice MRC benchmarks: RACE, DREAM and Cosmos QA, obtaining consistent and remarkable performance improvement with observable statistical significance level over strong baselines. Our code is available at https://github.com/Yilin1111/RekNet.

Index Terms—Natural Language Processing, Machine Reading Comprehension, Knowledge Enhancement, Reference Extraction, Reading Strategy.

I. INTRODUCTION

MACHINE reading comprehension (MRC) is a challenging natural language understanding task which requires machines to answer questions according to given passages [1], [2]. According to the formats of expectant answers [3], [4], MRC tasks can be roughly divided into generative task (to generate answer texts to given questions) [5], extractive task (to extract spans from given contexts for answer prediction) [6], multi-choice task (to select the most appropriate answer among given candidates) [7] and Yes/No task (to identify the authenticity of given declarative sentences, which can be regarded as a simplified version of multi-choice task) [8]. Among above tasks, multi-choice task requires model to have stronger ability for reading and comprehension due to potential gap between contexts and given candidates, which is the focus of this work.

In recent years, various datasets and tasks have been proposed, promoting a rapid improvement of MRC techniques [9], [10], [11]. To solve challenging MRC tasks, one of the most popular practice to is adopting powerful pre-trained contextualized models as encoder to obtain contextualized representations [12], [13], [14], [15]. Instead of better exploiting pre-trained encoders, this work is motivated by human reading and comprehension strategies to improve the performance of model. According to Dual Process Theory of cognition psychology [16], [17], [18], [19], [20], the cognitive process of human brains potentially involves two distinct types of procedures: contextualized perception (reading) and analytic cognition (comprehension), where the former gathers information in an implicit process, then the latter conducts the controlled reasoning and execute goals [21]. In further researches, Bengio [22] summarized two above cognitive sub-processes into more universal information processing systems: Implicit System and Explicit System, providing the paradigm architecture for machines to simulate human reading and comprehension strategies.
Inspired by above theories and strategies, we decouple MRC into \textit{sketchy reading} by extracting the critical spans from passages, and \textit{extensive reading} by quoting external knowledge explicitly. Furthermore, we propose a knowledge enhancement model based on extracted critical information, called \textit{RekNet (Reference Knowledgeable Network)}. In detail, \textit{RekNet} refines the fine-grained critical information by a span extraction module and defines it as \textit{Reference Span}, then quotes relevant explicit knowledge in the form of quadruples by the co-occurrence information of \textit{Reference Span} and candidate answers. One sample process of \textit{RekNet} is shown in Figure 1. After integrating the passage and enriched question (the integration of the original question and co-occurrence information of all candidate answers), we refine \textit{Reference Span} to obtain relevant knowledge quadruples for answer prediction.

The proposed \textit{RekNet} is evaluated on three multi-choice MRC benchmarks, RACE \cite{23}, DREAM \cite{24} and Cosmos QA \cite{25}, with significant and consistent performance improvement compared to baselines, and passes the significance test of MRC task \cite{26}. In summary, our main contributions are follows: i) We propose a novel reference-based knowledge enhancement model \textit{RekNet}, which makes the first attempt to integrate critical information extraction and knowledge injection for inference on MRC tasks. ii) \textit{RekNet} uses novel knowledge quadruples to quote relevant and credible knowledge, highlighting the effectiveness of quantitative knowledge items in multi-choice MRC tasks.

II. RELATED STUDIES

To solve multi-choice MRC tasks, existing studies mostly focus on more powerful pre-trained contextualized encoders \cite{13}, \cite{27}, \cite{28}, \cite{29}, or more complex network design to model the interactions between input sequences \cite{30}, \cite{31}, \cite{52}, \cite{53}. Instead, some studies attempt to improve the performance by human reading strategy simulation, among which \textit{knowledge injection} and \textit{reading strategy simulation} are in widespread attention.

A. Knowledge Injection

Observing the drawbacks of the lack of commonsense in MRC models, some researchers attempt to inject external knowledge explicitly \cite{43}, \cite{35}, \cite{36}. Xia \cite{37} used auxiliary tasks to obtain relevant knowledge, while Mihaylov \cite{38} encoded external commonsense knowledge as key-value memory under cloze-style setting. For non-MRC tasks, Chaudhuri \cite{39} quoted domain-specific knowledge with GRUs to refine and encode keywords, and Lin \cite{40} quoted a conceptual sub-graph of external knowledge to get better performance. Feng \cite{41} quoted multi-hop commonsense for simple questions without relevant contexts, while Lin \cite{42} proposed a constrained text generation task for generative commonsense reasoning. With the increasing negative impact of knowledge noise on model performance, researchers also explored to quote more relevant knowledge and reduce knowledge noise \cite{43}, \cite{44}. Though some of the above studies have noticed the importance of relevant knowledge selection, they generally ignore the powerful utility of critical information to knowledge injection as well as the filtration of untrustworthy knowledge.

B. Reading Strategy Simulation

Inspired by human reading strategies, some researchers design specific reading strategies for the model to improve MRC performance \cite{45}. Zhang \cite{26} proposed a retrospective reader for span-based MRC tasks, and Sun \cite{30} designed three specific human reading strategies for modeling enhancement. Among all reading strategies, \textit{evidence information extraction} has been paid great attention \cite{47}. Wang \cite{21} gave the first attempt to extract evidence sentences in single multi-choice MRC task, and Yadav \cite{48} applied this method to multi-hop QA tasks. Niu \cite{49} supervised evidence extractor with auto-generated evidence labels in an iterative process. However, all the above studies only execute information extraction at a more coarse-grained (sentence-level) granularity, lacking of sufficient accuracy compared to information extraction at fine-grained granularity, as well as adequate interaction between salient knowledge pieces.

C. Our Method

This work differs from previous studies as following: i) To highlight question-aware critical information from context, we model the \textit{Reference Span}, instead of using the whole context for inference or explicit knowledge retrieving; ii) To alleviate the negative impact of untrustworthy knowledge, our method quotes relevant explicit knowledge in the form of quadruple by adding confidence value of each knowledge item as an quantitative indicator over previous knowledge triplet.

To our best knowledge, our model is the first \textit{reference-based knowledge enhancement} model for multi-choice MRC tasks.

III. PRELIMINARY EXPERIMENTS

To explore general characteristics of multi-choice MRC tasks, we randomly extract 50 examples in DREAM, RACE and Cosmos QA respectively, finding that 38\%, 28\% and 34\% of the examples can be inferred \textbf{just} by several adjacent phrases in one single sentence, without explicit knowledge or logical calculation. This finding indicates that, in multi-choice MRC tasks, critical fine-grained information such as concise spans may contain the salient information for answer prediction directly, which can serve as an important reference indicator.

\begin{table}[h]
\centering
\caption{Preliminary Experiment 1: the exploration of overall utility of \textit{Reference Span}.}
\begin{tabular}{|l|c|c|}
\hline
\textbf{Model} & \textbf{Dev} & \textbf{Test} \\
\hline
Baseline (ALBERT\textsubscript{base}) & 65.74 & 65.36 \\
+ Reference Span & 67.65 & 67.86 \\
Reference Span only & 59.02 & 58.94 \\
\hline
\end{tabular}
\end{table}

Inspired by above finding, we directly replace the original passage with extracted \textit{Reference Span} as the input on DREAM, and the replaced model achieves acceptable performance as Table I shows. In detail, the \textit{Reference Span} is
extracted by another ALBERT\textsubscript{base} pre-trained on SQuAD 2.0, and the Baseline is fed with the triple of \{passage, question, candidate answers\} while Reference Span only is fed with \{Reference Span, question, candidate answers\}. + Reference Span represents the pooled logits of two above sequence triplets are concatenated for prediction, to both take advantage of the original redundant and refined critical information. From the results of Preliminary Experiment 1, there are two main factors which may cause the unsatisfactory performance according to our observation:

i) difficult questions that require external knowledge (as Figure [1] shows);

ii) potential mistakes from critical information extraction.

We find that 26\% examples of DREAM, 20\% examples of RACE and 62\% examples of Cosmos QA requires explicit knowledge for answer prediction in above 50 random examples, the first issue can be handled by augmentation from extra sources. Furthermore, the second issue can be alleviated by modeling both the original passage-aware and refined reference-aware sequences, as the superior result (+2.30\%) of + Reference Span showing in Table [1]

To further verify whether fine-grained information (i.e., several adjacent phrases in the preliminary finding) has more positive influence than coarse-grained information (i.e., evidence sentences in existing studies) for critical information extraction, we design Preliminary Experiment 2, where two more baselines obtain more coarse-grained Reference Sentences on DREAM. The first baseline calculates TF-IDF scores of each sentence in the passage and given question, denoting the largest appearing score as $s_{\text{max}}$. Then the model retains all sentences whose scores $s$ satisfying: $s \geq 0.7 \times s_{\text{max}}$ and splices them by spaces, as the input Reference Sentence. The other one uses our pre-trained ALBERT\textsubscript{base} (with the same setting to Preliminary Experiment 1) to obtain the whole sentence that contains Reference Span as Reference Sentence.

Table II. Preliminary Experiment 2: the exploration of fine-grained information.

| Method                  | Dev  | Test |
|-------------------------|------|------|
| TF-IDF Method           | 56.60| 55.44|
| ALBERT\textsubscript{base} Method (sentence) | 58.67| 58.35|
| ALBERT\textsubscript{base} Method | 59.02| 58.94|

As Table II shows, with Reference Span we extracted, the baseline obtains extra 0.47\% improvement on average comparing with general coarse-grained information extraction method, which proves the superiority of the fine-grained Reference Span.

Inspired by both our findings above and human reading and comprehension experiences [50], [26], we design our model by following a two-stage reading strategy, as critical information extraction and knowledge injection:

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

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.

IV. RekNet

Multi-choice MRC tasks can be defined as a triplet $(P, Q, A)$, where $P$ is passage, $Q$ is question with $n$ candidate answers: $A = \{A_1, ..., A_n\}$. Let $A_{\text{correct}} \in A$ be the correct answer of $Q$, the aim of multi-choice MRC model is to make:

\[
P(A_{\text{correct}} | P, Q, A) \geq P(A_i | P, Q, A),
\]

where $i \in (1, ..., n)$ and $P$ represents the probability of each candidate answer.

Another element $R$ is employed in our proposed RekNet, which is implemented as the critical information span (named as Reference Span) for each question. Furthermore, we combine the co-occurrence information of all candidate answers $A$ with the original question $Q$, generating enriched question $Q'$ for RekNet. Therefore, RekNet defines multi-choice MRC tasks to a quadruple $(P, Q', R, A)$ instead.

The overall framework of RekNet is shown in Figure 2 consisting of three modules: Reference Finder, Knowledge Adapter and Integrator. Triplet $(P, Q', A)$ is the input of RekNet, and the lower Integrator simulates human sketchy reading while the upper Integrator simulates human extensive reading. Details of above modules are shown in following subsections.

A. Reference Finder

RekNet employs Reference Finder to extracts Reference Span\footnote{In this work, RekNet only extracts one consequent Reference Span, which leads to the optimal performance in evaluated multi-choice MRC benchmarks.} and encodes it for critical information enhancement and relevant knowledge filtration. Reference Finder has two main components, called Reference Extractor and Reference Encoder, as Figure 3 shows.
• Enriched Question Q’
One challenge in multi-choice MRC task is the implicit critical information in given candidate answers. Without referring to candidate answers, it may be hard for model to extract precise critical information for the given question. Take Dialogue 1 in Table III for example, only model knows the key point to the given question is working situation of the man, it can refine the critical information accurately. Thus, for each question, we pick up the co-occurrence information of all candidate answers and add them to the head of the original question Q in order, forming the enriched question Q’. Above action facilitates Reference Finder to extract Reference Span more precisely.

Table III. Sample dialogue in DREAM for Q’, where bold texts in passage represent critical information for answer prediction while those in answer candidates represent co-occurrence information for the generation of Q’.

| Dialogue 1 | | Dialogue 2 | |
| --- | --- | --- | --- |
| ... | W: You worked for a large company before, didn’t you? | ... | ... |
| M: Yes, I did. But I prefer a small company. | | W: Is it really different? | |
| M: Oh, yes. It’s much different. I like a small company because it’s more exciting. | | M: Yes, I did. | |
| W: Is it really different? | | W: Is it really different? | |
| M: Oh, yes. It’s much different. I like a small company because it’s more exciting. | | M: Yes, I did. | |

In detail, for each candidate answer \( A_i \in A \), we construct a prototype token set \( A_i \) which contains the prototypes of all tokens in \( A_i \). We employ \( \text{spaCy} \)\(^3\) to execute lemmatization of each token in \( A_i \), and obtain all prototypes to constitute above prototype token set \( A_i \). Then we keep a co-occurrence token set \( T = \{t_1,...,t_c\} \), where \( c \) is the total of co-occurrence tokens, and for each co-occurrence token \( t_j \in T \) we have:

\[
t_j \in A_i, \quad \forall i \in \{1,...,n\}.
\]

We splice all prototype tokens in \( T \) with space in order and add the co-occurrence information with “.” into the head of Q to constitute Q’ ultimately.

• Reference Extractor

Reference Extractor to extract Reference Span. Following \([13]\), the input sequence of Reference Extractor is \([CLS] Q’ [SEP] P [SEP] \). We define the hidden size of the pre-trained contextualized encoder as \( H \) and the representation on the final hidden layer for the \( i \)-th input token as \( T_i \in \mathbb{R}^H \), then we score the predicted span starts from the \( i \)-th input token and ends at the \( j \)-th input token as \( S \cdot T_i + E \cdot T_j \), where \( S/E \in \mathbb{R}^H \) is the introduced start/end vector. Ultimately, span with the largest score of the former formula is chosen as the predicted Reference Span.

• Reference Encoder

RekNet employs the contextualized encoder in its baseline model as Reference Encoder, which shares all parameters to the Encoder in Figure 2. Reference Encoder encodes each \((Q’, A_i)\) pair with R in the form of \([CLS] R [SEP] Q’ + A_i [SEP] \) making RekNet obtain critical information enhancement from Reference Span directly; and encodes out a set of Reference Vectors \( RV = \{RV_1,...,RV_n\} \) where \( RV_i \in \mathbb{R}^H, i \in \{1,...,n\} \).

Similar to Reference Encoder, Encoder in Figure 2 takes \([CLS] P [SEP] Q’ + A_i [SEP] \) as its input sequence, and produces a similar set of Passage Vectors \( PV = \{PV_1,...,PV_n\} \), where \( PV_i \in \mathbb{R}^H, i \in \{1,...,n\} \). Ultimately, each element \( RV_i \) in \( RV \) will fuse with corresponding \( PV_i \) in the lower Integrator.

B. Knowledge Adapter

RekNet employs Knowledge Adapter to quote relevant explicit knowledge and encode knowledge quadruples, to highlight credible knowledge items. Figure 4 depicts the framework of Knowledge Finder and Knowledge Encoder.

• Knowledge Finder

For each input example, Knowledge Finder searches question-relevant knowledge items based on \((R, A)\) binary, due to the plentiful information and relatively concise length of \( R \). In detail, Knowledge Finder quotes knowledge items whose subject entity and object entity appear in the prototype list of \( R \) and \( A \) respectively. Above prototype matching is similar to the prototype matching for enriched question generation: we employ \( \text{spaCy} \)\(^4\) to execute lemmatization of each token in \( R \) and \( A \), and obtain prototypes to constitute prototype lists of \( \mathcal{R} \) and \( \mathcal{A} \). Besides, we also attempt cosine similarity for entity matching, getting no significant improvement compared with above simple method.

Knowledge items are saved in the form of quadruples \((\text{sub, rel, obj, con})\), which stands for the subject, relation, object and confidence value of each knowledge item. (\text{doctor, can, help_sick_person}, 4.472) is a sample quadruple, where the subject, relation, and object can be a word or a phrase, and confidence value is a number larger than 0.1. Under this setting, RekNet can effectively separate knowledge quadruples with low confidence values from blank knowledge quadruples. Larger confidence value indicates the knowledge item is more

\(^3\)\(\text{spaCy}\) denotes string splicing operation by one space.\(^4\)\(RV_i\) is the embedding vector of \([CLS]\) from the last hidden layer of Reference Encoder.
Then RekNet sends all \( m = k \times n \) embeddings to a weighted attention module with their confidence values, and the main operation can be expressed as:

\[
KV_i = \sum_{j=1}^{k \times n} \text{Softmax}(\text{WeiAtt}(RV_i, h_j, c_j))^T h_j, \quad i \in \{1, \ldots, n\},
\]

where \( h_j \) and \( c_j \) are hidden_obj and confidence value of the \( j \)-th knowledge quadruple, \( KV_i \) and \( RV_i \) are respectively Knowledge Vector and Reference Vector of the \((R, A_i)\) binary. The weighted attention module \text{WeiAtt} is employed to compute the unnormalized contribution proportion for each knowledge quadruple, which can be formulated as:

\[
\text{WeiAtt}(RV_i, h_j, c_j) = (RV_i^T Wh_j) \cdot c_j, \quad j \in \{1, \ldots, m\},
\]

where \( W \) is a linear transform matrix, and \( \cdot \) is scalar multiplication.

C. Integrator & Classifier

- **Integrator**
  
  There are two Integrators in RekNet to integrate information from two input embedding representations to one fusion representation. The structure of Integrator is illustrated in Figure 5.

- **Classifier**
  
  For classifier, RekNet employs a linear softmax layer to calculate score for each candidate answer. Furthermore, we use standard Cross Entropy Loss as the loss function. Ultimately, there is a Dropout Layer to prevent over-fitting.

V. Experiments

A. Setup

We run the experiments on 8 NVIDIA Tesla P40 GPUs, and the implementation of RekNet is based on the Pytorch implementation of ALBERT_xxlarge [27]. In the experiments, we adopt ALBERT pre-trained on SQuAD 2.0 [52] as Reference Extractor, due to SQuAD 2.0 has the largest contribution to other MRC tasks among extractive tasks [4]. Besides, we eliminate the possibility of extracting null Reference Span.

---

**Fig. 4.** The framework of Knowledge Adapter. \( \cdot \) stands for scalar multiplication operation, and \( m = k \times n \) is the number of knowledge quadruples. \( K_j \) refers to the \( j \)-th knowledge quadruple, \( C_j \) and \( h_j \) refers to the confidence value and embedding representation of \( K_j \).

**Fig. 5.** The framework of Integrator, where all vectors are in the size of \( \mathbb{R}^H \).

creditable, and we obtain confidence values from the weight values in ConceptNet [51] then normalize them into \([0, 1, 2.0]\), by setting the confidence values larger than \(2.0\) to 2.0 due to their minor scale and uneven distribution. One typical weight value in the original ConceptNet is 1.0, but the value can be higher or lower, and 99.37% of the weight values locate in \([0.1, 2.0]\) in the original ConceptNet.

RekNet sets each input example can only retain \( k \times n \) knowledge quadruples with the largest confidence values at most, where \( k \) is the number of quotable knowledge quadruples for each candidate answer, and this design can reduce knowledge noise significantly. If one example can not get enough quadruples, RekNet will add blank quadruples whose confidence values are 0 to fill up to the specified amount.

**Knowledge Encoder**

We define the remaining part of Knowledge Adapter as Knowledge Encoder. For each \((R, A_i)\) binary, the input embedding for each word or phrase in knowledge quadruples is sent to a series of GRU modules to encode the quadruple as following:

\[
\begin{align*}
\text{hidden}_{\text{sub}} &= \text{GRU}(\text{emb}_{\text{sub}}, 0), \\
\text{hidden}_{\text{rel}} &= \text{GRU}(\text{emb}_{\text{rel}}, \text{hidden}_{\text{sub}}), \\
\text{hidden}_{\text{obj}} &= \text{GRU}(\text{emb}_{\text{obj}}, \text{hidden}_{\text{rel}}),
\end{align*}
\]

where \( \text{emb}_{\text{sub}}, \text{emb}_{\text{rel}}, \text{emb}_{\text{obj}} \) are the input embeddings of subject, relation and object, and \( \text{hidden}_{\text{sub}}, \text{hidden}_{\text{rel}}, \text{hidden}_{\text{obj}} \) are the last hidden representations of GRU. With above encoding design, the directionality of input quadruples can be retained, and the embedding of explicit knowledge is in the same vector space as the plain tokens.
to ensure all examples can get a non-null Reference Span, because there exist questions with null span as the golden answer in SQuAD 2.0. We set the number of knowledge quadruples for each candidate answer k to 5, which leads to the best performance for RekNet. The fine-tuning hyper-parameters of RekNet are given in Table IV which leads to the optimal performance. In detail, statistics about the length are based on token level, and Source Content denotes the input sequence of model \((CLS)\ [SEP]\ Q' + A_i [SEP])

Table IV. The fine-tuning hyper-parameters of RekNet.

| Hyper-parameter | RACE | DREAM | Cosmos QA |
|-----------------|------|-------|-----------|
| Learning Rate   | 1e-5 | 1e-5  | 1e-5      |
| Batch Size      | 32   | 24    | 32        |
| Warmup Steps    | 1000 | 50    | 1000      |
| Maximum Sequence Length | 384 | 512   | 384       |
| Maximum Reference Span Length | 256 | 256   | 256       |
| Training Epochs | 2    | 2     | 5         |
| Steps to Save Checkpoints | 3000 | 400   | 2000      |
| Maximum Source Content Length | - | 1480 | - |
| Average Source Content Length | - | - | 396 |

B. Dataset

We employ RACE [23], DREAM [24] and Cosmos QA [25] as our evaluation benchmarks, and ConceptNet 5.7.0 [51] as external commonsense knowledge source. The details of above datasets are shown as following:

**RACE** is a large-scale MRC task collected from English examinations, which contains nearly 100,000 questions. The passages are in the form of articles with diversified topic domains, and most questions require contextual reasoning. Each question in RACE has 4 candidate answers.

**DREAM** is a dialogue-based dataset for multi-choice MRC, containing more than 10,000 questions. More than 80% of the questions in DREAM are non-extractive and require reasoning from multi-turn dialogues, and more than one third of the given questions involve commonsense knowledge. Each question in DREAM has 3 candidate answers.

**Cosmos QA** is a large-scale MRC task collected from people’s daily narratives, which has about 35,600 questions. The questions are about the causes or effects of events, which require evidence extraction as well as knowledge injection. Each question in Cosmos QA has 4 candidate answers.

**Knowledge Source** ConceptNet 5.7.0 [51] is one of the largest structured knowledge base with confidence weight for each knowledge item as its knowledge source. Due to the huge size of ConceptNet, we only retain its English concepts. Following Lin [40], we merge some relations which share similar meanings, and the retained relations are listed in Table V. To obtain the input embedding for each word or phrase in ConceptNet, we adapt the pre-trained 100-dimension GloVe [53] embedding vectors.

C. Results

We adopt accuracy as evaluation criteria for multi-choice MRC. Tables VI-VIII show the results of RekNet compared with its baselines, as well as other public models on the leaderboard. Though some public works may have better performance with larger model size (e.g., 3.9 billion parameters for Megatron-BERT, 11 billion parameters for T5) or special complex neural network (like DUMA), our proposed RekNet obtains consistent improvements over all evaluated benchmarks, with acceptable model size. For quantitative proof, significant test [26] shows that RekNet improves its baseline on evaluated benchmarks significantly with \(p\text{-value} < 0.01\).

Table V. Retained relations in ConceptNet.

| Retained relations in ConceptNet |
|----------------------------------|
| locate, can, causes, product, desires, antonym, situation, is, entails, isa, disable, unnecessary, have, relatedto, field, capital, influence, occupation, language, leader |

Table VI. Results on DREAM. Results in the first domain are from the leaderboard. SC denotes single choice and TL denotes transfer learning.

| Model                     | Dev | Test |
|---------------------------|-----|------|
| FTLM++ [24]               | 58.1| 58.2 |
| BERT\text{\textsubscript{base}} [13] | 63.4| 63.2 |
| BERT\text{\textsubscript{large}} [13] | 66.0| 66.8 |
| XLNet\text{\textsubscript{base}} [28] | - | 72.0 |
| RoBERTa\text{\textsubscript{large}} [54] | 85.4| 85.0 |
| RoBERTa\text{\textsubscript{large}} + MMM [55] | 88.0| 88.9 |
| ALBERT\text{\textsubscript{xxlarge}} + DUMA \text{\textsubscript{base}} [21] | 89.9| 90.4 |
| ALBERT\text{\textsubscript{xxlarge}} + DUMA + MTL | - | 91.8 |
| ALBERT\text{\textsubscript{base}} (rerun) | 65.7| 65.6 |
| RekNet on ALBERT\text{\textsubscript{base}} | 68.9| 68.7 |
| RekNet on ALBERT\text{\textsubscript{xxlarge}} (rerun) | 89.2| 88.5 |
| RekNet on ALBERT\text{\textsubscript{xxlarge}} | 89.8| 89.6 |

Table VII. Results on RACE. Results in the first domain are from the leaderboard. SC denotes single choice and TL denotes transfer learning.

| Model                     | Dev (M / H) | Test (M / H) |
|---------------------------|-------------|--------------|
| BERT\text{\textsubscript{base}} [13] | 64.6 (-/–) | 65.0 (71.1 / 62.3) |
| BERT\text{\textsubscript{large}} [13] | 72.7 (76.7 / 71.0) | 72.0 (76.9 / 70.1) |
| XLNet\text{\textsubscript{base}} [28] | 80.1 (-/–) | 81.8 (85.5 / 80.2) |
| XLNet\text{\textsubscript{base}} + DCMN+ [56] | - (-/–) | 82.8 (86.5 / 81.3) |
| RoBERTa\text{\textsubscript{large}} [54] | - (-/–) | 83.2 (86.5 / 81.8) |
| RoBERTa\text{\textsubscript{large}} + MMM [55] | - (-/–) | 85.0 (89.1 / 83.3) |
| TS-11B [19] | - (-/–) | 87.1 (-/–) |
| ALBERT\text{\textsubscript{xxlarge}} + DUMA [21] | 88.1 (-/–) | 88.0 (90.9 / 86.7) |
| TS-11B + UnifiedQA [4] | - (-/–) | 89.4 (-/–) |
| Megatron-BERT-3.9B [53] | - (-/–) | 89.5 (91.8 / 88.6) |
| ALBERT\text{\textsubscript{xxlarge}} +SC +TL [59] | - (-/–) | 90.7 (92.8 / 89.8) |
| ALBERT\text{\textsubscript{base}} (rerun) | 67.9 (72.3 / 66.7) | 67.2 (72.1 / 65.2) |
| RekNet on ALBERT\text{\textsubscript{base}} | 70.7 (73.9 / 68.9) | 70.3 (74.4 / 68.3) |
| RekNet on ALBERT\text{\textsubscript{xxlarge}} (rerun) | 86.6 (89.4 / 85.2) | 86.5 (89.2 / 85.4) |
| RekNet on ALBERT\text{\textsubscript{xxlarge}} | 87.8 (91.1 / 86.4) | 87.8 (90.1 / 86.8) |

Through further observation, we find the improvement of RekNet becomes less as the size of baseline increases (e.g., 3.1% vs. 1.1% on DREAM). By analyzing 50 randomly selected error cases over the baselines on DREAM, the proportion of the cases can be inferred just by several adjacent phrases decreases from 38% to 26% on ALBERT\text{\textsubscript{base}} and 16% on ALBERT\text{\textsubscript{xxlarge}}, and the cases requiring external knowledge decreases from 26% to 22% on ALBERT\text{\textsubscript{base}} and 16% on ALBERT\text{\textsubscript{xxlarge}}. This statistics indicates that, with more parameters and stronger encoding ability, larger model might have stronger ability to capture the relationships among
Table VIII. Results on Cosmos QA. Results in the first domain are from the leaderboard, and only public works are listed.

| Model                      | Dev  | Test  |
|----------------------------|------|-------|
| BERT\textsubscript{base}   | 66.2 | 67.1  |
| RoBERT\textsubscript{large} | 81.7 | 83.5  |
| RoBERT\textsubscript{large} + CEGI \cite{60} | 83.8 | 83.6  |
| ALBERT\textsubscript{xxlarge} + GDIN \cite{61} | 83.6 | 84.6  |
| RoBERT\textsubscript{large} + ALICE \cite{62} | 83.6 | 84.6  |
| T5-11B \cite{57}          | –    | 90.3  |
| T5-11B + UNICORN \cite{63} | –    | 91.8  |
| ALBERT\textsubscript{base} (rerun) | 63.1 | 63.7  |
| RekNet on ALBERT\textsubscript{base} | 65.7 | 65.8  |
| ALBERT\textsubscript{xxlarge} (rerun) | 85.0 | 84.8  |
| RekNet on ALBERT\textsubscript{xxlarge} | 85.9 | 85.7  |

adjacent phrases as well as encode knowledge implicitly by itself, which weakens the original benefit of critical information extraction and knowledge injection.

Table IX. Training parameters in RekNet and partial public works in Tables VII–VIII. The rest works do not provide the number of parameters in their models.

| Model                  | Parameters |
|------------------------|------------|
| ALBERT\textsubscript{base} \cite{27} | 12M        |
| ALBERT\textsubscript{base} + DUMA \cite{31} | 13.5M      |
| ALBERT\textsubscript{base} + DCMN+ \cite{59} | 19.4M      |
| BERT\textsubscript{base} \cite{13} | 108M       |
| ALBERT\textsubscript{xxlarge} \cite{27} | 235M       |
| ALBERT\textsubscript{xxlarge} + DUMA \cite{31} | 292M       |
| BERT\textsubscript{large} \cite{13} | 334M       |
| XLNet\textsubscript{large} \cite{28} | 345M       |
| RoBERT\textsubscript{large} \cite{54} | 355M       |
| Megatron-BERT-3.9B \cite{58} | 3.9B       |
| T5-11B \cite{57} | 11B        |
| T5-11B + UnifiedQA \cite{4} | 11B        |
| T5-11B + UNICORN \cite{63} | 11B        |
| ALBERT\textsubscript{base} (rerun) | 11.2M      |
| RekNet on ALBERT\textsubscript{base} | 13.5M      |
| ALBERT\textsubscript{xxlarge} (rerun) | 244M       |
| RekNet on ALBERT\textsubscript{xxlarge} | 277M       |

Furthermore, we collect the training parameters of RekNet and partial public works in Tables VII–VIII as Table IX shows. In conclusion, compared with the 244M parameters in the baseline ALBERT\textsubscript{xxlarge}, only approximately 13.5% additional parameters are introduced into RekNet, which is less than other comparable works \cite{31, 59}, as well as demonstrates the conciseness and efficiency of RekNet.

VI. ANALYSIS

A. Ablation Studies

As Figure 2 shows, there are two main modules (Reference Finder and Knowledge Adapter) in RekNet. To explore the necessity of each module, we remove one of them for each time and keep the hyper-parameters unchanged, obtaining the results of ablation studies on DREAM in Table X.

Above results show that, removing Reference Finder brings 1.07% average performance reduction to the intact model and removing Knowledge Adapter brings 1.05% performance reduction, indicating both of proposed modules are indispensable to RekNet. This finding further demonstrates the rationality of the reading strategy RekNet follows:

i) Without Reference Finder, Knowledge Adapter executes extensive reading blindly in full-text range, and irrelevant knowledge noise may be quoted into RekNet, making it suffer from lacking reading process. However, with the well-designed knowledge structure (quadruple) as well as the numeric restriction, RekNet can still get relatively positive improvement by quoting explicit knowledge.

ii) Without Knowledge Adapter, RekNet cannot introduce explicit knowledge separated from the context, leading to the lack of relevant external information and cannot reflect human comprehension process.

To study the influence of the fusion order, we exchange the order of $RV_i$ and $KV_i$ integrates with $PV_i$, which could be considered to exchange the positions of two Integrators. As Table X shows, relocation of these two modules brings negative impact to RekNet, which may be because:

i) The context in Reference Span has a higher degree of similarity to the initial context in passage, and knowledge quadruples are extracted according to Reference Span. Integration of two representations of more similar contexts can makes model learn more features than combining two more different ones.

ii) Relocation RekNet violates the order of natural human reading comprehension strategy, which may quote and over-analyze irrelevant knowledge items.

B. Studies on Enriched Question $Q'$

Table XI. Statistics on Experimental Datasets for $Q'$.

| Dataset       | DREAM  | RACE  | Cosmos QA |
|---------------|--------|-------|-----------|
| Co-occurrence Proportion (%) | 62.8   | 55.8  | 48.1      |
| Average Length of Added Tokens | 1.91   | 2.64  | 2.44      |

To show the advantages of $Q'$ compared with $Q$, we show more detailed examples in DREAM and RACE in Table XII. We present comprehensive statistics on the proportion of co-occurrence examples on all experimental datasets, together with the average length of added tokens, as Table XI shows. Most original questions will be enriched by co-occurrence information with brief but critical words according to the limited length of added tokens, avoiding excessive redundant words, such as Dialogue 2 and Document 1 in Table XII.

Furthermore, to learn the improvement brought by enriched information, we design a degradation experiment on DREAM, by degenerate $Q'$ to $Q$ in RekNet. Results in Table XIII indicate that $Q'$ can improve the performance of MRC model and the main contribution of $Q'$ is to help model get more precise Reference Spans for multi-choice MRC tasks. Besides, to expand RekNet to other MRC categories such as extractive
Table XII. More sample dialogues and documents in DREAM and RACE for Q’, where bold texts in passage represent critical information for answer prediction while those in answer candidates represent co-occurrence information for the generation of Q’.

| Dialogue 2 |
|------------|
| ... M: We’ll take the two rooms. 
W: Very good, sir. Would you please register? Write your name and address on this card. Thank you. Is this your luggage? 
M: Yes. We have four suitcases. 
W: All right. The bellboy will bring them up for you. You will be in rooms 403 and 405. How long do you plan to stay in Boston? ... |

Q: The man has to ____.
A. register by writing his name and address. (correct)
B. register for his suitcases.
C. register for the tour arrangement.
Q’? Register. The man has to ____.

... But early-airport people get ulcers, heart attacks and bite their fingernails to the bone. Late-airport people almost don’t realize they are flying. A guy of that kind once said, don’t hurry. If you miss your flight, it’s because God doesn’t want you to go. ...

Q: We can learn from the passage that ____.
A. late-airport persons are often nervous.
B. early-airport persons are always at ease during the flight.
C. early-airport persons get their baggage first after the landing.
D. late-airport persons always take things easy. (correct)
Q’? Airport person. We can learn from the passage that ____.

MRC, Q’ can be degenerated into Q, with acceptable performance degradation.

Table XIII. Results of question degradation experiments.

| Q or Q’ | Dev  | Test |
|---------|------|------|
| Q’ to All Modules | 68.87 | 68.74 |
| Q to All Modules | 67.60 | 68.12 |
| Q’ to Reference Finder and Q to Other Modules | 68.46 | 68.50 |

C. Studies on Reference Span

In this section, we conduct a series of experiments to explore the effect of Reference Span on RekNet, in terms of quality, quantity and integration method.

i) Quality of Reference Span. Since the framework of RekNet employs a two-stage design (Reference Span extraction at the first stage and knowledge quadruple injection at the second stage), one natural concern is that, whether the quality of Reference Span has a significant impact on the ultimate performance of RekNet, especially when the quality of Reference Span is relatively low. Therefore, we design three baseline models with different Reference Spans in lower quality than the original RekNet adopted:

- **TF-IDF Reference Span**: A heuristic TF-IDF method is employed to extract Reference Span, instead of the pre-trained Reference Extractor. In detail, we divide the original context into clauses and then calculate the TF-IDF similarity score of each clause and enriched question Q’.

The clause with the highest score is extracted as the **TF-IDF Reference Span**.

- **Attention Reference Span**: We replace the Reference Extractor by a simple layer of attention calculation. In detail, we employ an untrained contextualized encoder ALBERT base to encode each clause and enriched question Q’, then computes the attention score of the embedding of Q’ and each clause:

\[
 s_i = \exp\left(\sum_{j=1}^{N_{cla}} \text{emb}_{cla}(j) \cdot \text{emb}_{que}(i) \right), \quad i \in \{1, \ldots, N_{cla}\},
\]

where \( \text{emb}_{cla}, \text{emb}_{que} \) are the embeddings of the i-th clause and enriched question Q’. \( N_{cla} \) is the number of clause, and \( s_i \) is the attention score of the i-th clause. Then the clause with the highest score \( s_i \) is extracted as the **Attention Reference Span**.

- **Weak Reference Span**: For the low-quality Reference Span in this baseline, we pre-train the Reference Extractor on SQuAD 2.0 with only one training epoch to make the pre-training process inadequate and keep other processes and settings unchanged.

ii) Quantity of Reference Span. To study whether extraction of multiple Reference Spans can bring positive benefits to RekNet, we design a baseline which lets Reference Extractor extract multiple Reference Spans, compared with only one single Reference Span in proposed RekNet. First, we set the probability score of the Reference Span in the original RekNet as \( s_{max} \), which is scored by Reference Extractor. Then in RekNet with multiple Reference Spans, we modify Reference Extractor to retain all Reference Spans whose probability scores \( s \) satisfying \( s \geq 0.7 \times s_{max} \), to build up Reference Span Set. Ultimately, we remove overlapping sub-spans and splice all elements in Reference Span Set with “; “, and replace the original Reference Span with it.

iii) Integration Method of Reference Span. In our experiments, for each 1,000 training steps, the baseline ALBERT costs 134/1,267 seconds on base and xxlarge size respectively, while the original RekNet costs 196/2,063 seconds respectively, with 54.5% additional training time cost on average. Though the proportion of additional training time cost is not at a low level like additional parameters, we find that most of the additional training cost (more than 70%) comes from the encoding of Reference Span.

Thus, we explore one simplified integration method of Reference Span to simplify the integration of Passage Vector PV and Reference Vector RV, and improve the overall training speed of RekNet. In detail, we design an additional feature encoding layer in the Encoder in Figure 2 to encode the original context with Reference Span as one additional indicating feature. Since the integrated embedding \( E \) encodes both the original passage and Reference Span, we assign \( E \) to both \( PV \) and \( RV \) in RekNet, and remove Reference Encoder in this baseline.

7Following single sentence setting in [13], the input sequences are “[CLS] \( e_{cla} \) ” and “[CLS] \( Q \)” respectively, and \( \text{emb}_{cla}, \text{emb}_{que} \) are the embedding vectors of [CLS] token.

8As a result, the performance of Exact Match (EM) on SQuAD 2.0 drops from 79.21 (2 training epochs, for the original Reference Extractor) to 73.17 (1 training epoch).
For the detailed design of above simplified integration method, the integrated embedding in above baseline is the normalized sum of the original Contextualized Embedding and proposed Reference Embedding. Following the basic design in embedding layers of BERT-style models, the original Contextualized Embedding consists of Token Embedding $E_t$, Segmentation Embedding $E_s$, and Position Embedding $E_p$. For Reference Embedding $E_{Ref}$, we implement another embedding layer, which has 1 (for tokens in Reference Span) and 0 (for other tokens) two possible assignment values. The proposed Reference Embedding has the same embedding size as Contextualized Embedding, guaranteeing all above indicating embeddings are in the same vector space. Formulatively, the integrated embedding $E$ can be represented as:

$$E = \text{Norm}(E_t + E_s + E_p + E_{Ref}),$$

where $\text{Norm}()$ is a layer normalization function [64].

Table XIV. Results of RekNet and baselines about Reference Span in base size on DREAM.

| Model                   | Dev  | Test  |
|-------------------------|------|-------|
| Baseline (ALBERTbase)   | 65.74| 65.56 |
| RekNet                  | 68.74| 68.74 |
| + TF-IDF Reference Finder | 68.03 | 68.19 |
| + Attention Reference Finder | 68.39 | 68.35 |
| + Weak Reference Finder | 68.63 | 68.48 |
| + Multiple Reference Finder | 68.61 | 68.25 |
| + Simplified Integration | 68.45 | 68.40 |

Retaining all other settings the same, we evaluate RekNet and above baselines about Reference Span in base size on DREAM, and the results are shown in Table XIV which indicate that:

i) **Quality of Reference Span.** Though the overall performance of RekNet with a low-quality Reference Span is lower than the original RekNet, the slight degree of reduction illustrates that, RekNet does not suffer a lot from low-quality Reference Span like most two-stage framework models. RekNet possesses satisfactory robustness, especially when the extraction of Reference Span is not completely accurate.

The robustness may benefit from the integration with Passage Vector $PV$, which can be seen as a protective measure to the low-quality Reference Span. Besides, knowledge quadruple injection at the second stage does not have a strict requirement on the accuracy of Reference Span, since the Reference Span only serves as a guard for the contribution proportion of knowledge quadruples. One rough or even inaccurate range with less computation cost for Reference Span can still work well for the overall performance of RekNet.

ii) **Quantity of Reference Span.** We find the performance of RekNet gets a slight negative impact when it extracts multiple Reference Spans. The possible reasons may be, with multiple Reference Spans, model will distract attention to more irrelevant Reference Spans. Furthermore, more knowledge quadruples with weak correlation (in weak-relevant Reference Spans) will be quoted into RekNet, leading to further negative impact.

iii) **Integration Method of Reference Span.** With above simplified integration method, though the overall performance of RekNet drops slightly (0.38% on base size on average), the training time of RekNet reduces to 147/1,542 seconds on base and xxlarge size for 1,000 training steps, with 15.7% additional cost on average. Thus, we recommend RekNet with this simplified integration method to the researchers who pursue higher training or inference speed.

D. Analysis of Knowledge Quadruples

Instead of knowledge triplets in existing studies [38], [40], [41], RekNet quotes and encodes explicit knowledge quantitatively in the form of quadruples and achieves significant improvement. To prove the effectiveness of knowledge quadruples we proposed, we degenerate knowledge quadruples to knowledge triplets 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 $\mathbb{R}^{k \times 5}$ for each $(R, A)$ binary. We denote Mask Vector as $MV$, where $MV_i$ is the $i$-th element of $MV$. Then we set:

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

for Mask Vector, where $\beta$ is a non-negative threshold.

Table XV. Result of knowledge degeneration experiments.

| Model                  | Dev    | Test    |
|------------------------|--------|---------|
| Baseline (ALBERTbase)  | 65.74  | 65.56   |
| RekNet                 | 68.74  | 68.74   |
| Masked RekNet ($\beta = 0$) | 68.28  | 68.05   |

We first set $\beta = 0$ to degenerate knowledge quadruples into triplets, which treats all knowledge items equally. Result in Table XV indicates that, Mask Vector performs worse than the intact RekNet with confidence values, because there exists a huge amount of untrustworthy knowledge items in knowledge graph (ConceptNet), such as (abdomen, relatedto, thorax, 0.102). It is unreasonable to give them equal treatment to trustworthy ones.

![Fig. 6. The accuracy of Masked RekNet with different thresholds ($\beta$) on development set. KA: Knowledge Adapter.](image-url)
Then we implemented masked experiments over different $\beta$ on the development set of DREAM, and the results are shown in Figure 6. The figure shows that, among different thresholds, Masked RekNet performs better than RekNet without Knowledge Adapter (no knowledge), but worse than the intact RekNet (with knowledge quadruples).

Furthermore, the reasons for the trend of model performance with $\beta$ may be that, with lower $\beta$, model cannot filter out untrustworthy knowledge items effectively, while with higher $\beta$ model may filter out some credible and important knowledge. In summary, the results of above masked experiments prove the effectiveness of confidence values.

To show the effectiveness of knowledge filtering method of RekNet (using knowledge quadruples) in a visual way, we extract the final contribution proportion of each knowledge item for the example in Figure 1 and display the visualized proportions in Figure 7, together with the proportions in Masked RekNet (using mainstream knowledge triplets).

![Visual Contribution Proportion](image)

Fig. 7. The visual contribution proportion of each knowledge item for the dialogue in Figure 1. In this example, red, yellow and blue knowledge items are untrustworthy, irrelevant and golden ones, respectively.

As the figure depicts, by adopting knowledge quadruples, RekNet gives the lowest proportion to (lunch, relatedto, morning, 0.449) because it is untrustworthy, and gives (morning, relatedto, break, 0.872) a relatively low proportion in spite of its high confidence value, because it has less contextual relevance to the Reference Span. Compared with the intact RekNet, Masked RekNet adopting knowledge triplets may unnecessarily pay attention to untrustworthy knowledge items and less attention to precise ones, leading to wrong predictions.

### E. Statistics about Retrived Knowledge

In order to provide researchers with more useful information about the utilization of external knowledge in multi-choice MRC tasks, we make a brief statistics about the retrieved knowledge by RekNet to explore:

i) the overall demand of questions in multi-choice MRC tasks;

ii) which type of knowledge contributes more to multi-choice MRC tasks, or tends to attract more attention from proposed model.

As a result, we find 77.82%, 72.55% and 84.73% questions in DREAM, RACE and CosmosQA are covered by knowledge respectively, and each covered $(Q, A_i)$ pair has 2.42, 3.90, 2.69 knowledge items on average (3.08 quadruples in total). And for all questions (including uncovered questions), each question retrieves 9.84 knowledge quadruples on average (2.49 quadruples for each $(Q, A_i)$ pair).

Above statistics proves the rationality of our setting of $k = 5$ (the number of knowledge quadruples for each candidate answer) in the setup of experiments. Due to each $(Q, A_i)$ pair covered by knowledge retrieves approximately 3 pieces of knowledge items on average (nearly 4 pieces for RACE), smaller $k$ may 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 average performance on ALBERT$_{base}$ drops by 0.24% and 0.19% on RACE compared with $k = 5$.

As for the relations in knowledge quadruples, we have 20 types of different relations in total, and top 3 relationships in the original ConceptNet are: “relatedto” (67.65%), “isa” (17.48%) and “situation” (7.12%), but the top 3 relationships retrieved by RekNet are “relatedto” (72.13%), “antonym” (7.56%) and “locate” (8.54%). This finding indicates that:

i) RekNet tends to retrieve “relatedto” relationship due to this type of knowledge items accounts for more than two thirds knowledge items in the original ConceptNet;

ii) Knowledge item which indicates negative meaning (antonym: 0.72% → 7.56%) or is hard to learn by contextualized encoder (locate: 0.90% → 5.84%) is helpful for RekNet, instead of similar entities which can be easily learned by models (isa: 17.48% → 5.68%).

### F. Expand on Extractive MRC

In addition to the multi-choice MRC tasks focused on by this work, we further expand RekNet to other MRC categories such as extractive MRC tasks, to verify the versatility of the proposed methods in RekNet.

Following [13], the input sequence for extractive MRC can be represented as $[CLS] \ Q \ [SEP] \ P \ [SEP]$. We define the representation on the final hidden layer for the $i$-th input token as $T_i \in \mathbb{R}^H$, then we score the answer span starts from the $i$-th input token and ends at the $j$-th input token as $S \cdot T_i + E \cdot T_j$, where $S/E \in \mathbb{R}^H$ is the introduced start/end vector. Ultimately, span with the largest score of the former formula is chosen as the predicted answer span. Denoting $SP_i = \sum_j e^{s \cdot T_j}$ as the probability of the $i$-th token being the start of the answer span and $EP_i = \sum_j e^{e \cdot T_j}$ as the probability of being the end of the answer span, our training object is the sum of Cross Entropy Losses for the start and end token probabilities.

And for the implementation of RekNet on extractive MRC, we design the following adjustments to bridge the gap from multi-choice MRC:

i) As we mention in Section VI.B, we degenerate $Q'$ into $Q$ for the input sequences of RekNet, due to no candidate answer being given.

ii) For the representation alignment of each token in extractive MRC task, we remove Reference Encoder module and
adopt the simplified integration method (indicator embedding layer for Reference Span, mentioned in Section VLC) to integrate the given passage and extracted Reference Span.

iii) Knowledge Vector Integrator searches question-relevant knowledge items based on \((R, Q)\) binary instead of \((R, A)\) binary in multi-choice MRC, and we set the maximum number \(m\) of knowledge quadruples as \(m = k = 5\).

iv) We enhance the embedding vector of each input token by the integration of Knowledge Vector. Similar to the Integrator in Section IV.C, we execute concatenation and dimensionality reduction operations for them.

We evaluate above adjusted RekNet on ALBERT\_base on the development set of SQuAD 1.1 \[65\], and provide an example to show how RekNet works on extractive MRC tasks in Table \[XVII\]. Corresponding ablation experiment results are shown in Table \[XVI\].

### Table XVI. Sample document in SQuAD 1.1, where bold texts represent Reference Span and knowledge quadruples are in the form of "(subject, relation, object, confidence value) → contribution proportion".

| Document 2 |
|-------------|
| ... Fox paid for Deadpool, X-Men: Apocalypse, Independence Day: Resurgence and Eddie the Eagle, Lionsgate paid for Gods of Egypt, Paramount paid for Teenage Mutant Ninja Turtles: Out of the Shadows and 10 Cloverfield Lane, ... |

| Knowledge Quadruples: | (ind: independence, rel: relatedto, subj: july, value: 0.591) → 0.5697 |
|----------------------|------------------------------------------------------------------|
| (ind: independence, rel: relatedto, subj: july, value: 1.297) → 0.2688 |
| (ind: holiday, isa: holiday, value: 1.000) → 0.1501 |
| (holiday, isa: holiday, subj: day, value: 2.000) → 0.0103, (x: Isa, value: 1.000) → 0.0011 |

A: Independence Day: Resurgence (correct answer)

### Table XVII. Ablation experiment results of RekNet for extractive MRC.

| Model                  | EM  | F1  |
|-----------------------|-----|-----|
| Baseline (ALBERT\_base) | 82.66 | 89.91 |
| Adjusted RekNet        | 84.03 | 90.98 |
| - Reference Finder     | 82.97 | 90.42 |
| - Knowledge Adapter    | 83.58 | 90.69 |

As Document 2 in Table \[XVI\] shows, in the adjusted RekNet, Reference Finder extracts one less precise span as Reference Span (e.g., the bold texts in Document 2). In the above example, all movies paid by Fox have been extracted coarsely due to the lack of commonsense about “July Fourth holiday” in the question. Then Knowledge Adapter quotes related knowledge quadruples and scores them with final contribution proportions, highlighting critical information “Independence Day” in the Reference Span. With the integration of the above information, our adjusted RekNet ultimately enhances the embedding vector of each token and predicts the precise answer span.

Furthermore, as Table \[XVII\] shows, though some benefit components or methods are discarded for extractive MRC (such as enriched question \(Q'\) and Reference Encoder), the improvements in above ablation results prove that, RekNet possesses satisfactory versatility on diverse MRC categories such as extractive MRC. Besides, the limited contribution of Knowledge Adapter on extractive MRC indicates that, in most cases extractive MRC may benefit less from external knowledge injection than multi-choice MRC, which may be one main reason that most existing researches about knowledge enhancement \[36\], \[37\], \[34\], \[40\], \[41\], \[66\] as well as this work focus on multi-choice MRC tasks, instead of other MRC categories.

### G. Error Case Analysis

Table XVIII. Sample dialogue in DREAM requiring numeral calculation, where bold texts represent critical information which needs numerical calculation for answer prediction.

| Dialogue 3 |
|------------|
| ... W: The price for one person for a ten-day tour is only $1,088, which includes round-trip airfare. M: That sounds reasonable. By the way, do you have a discount for two? W: Yes, you can have a 10% discount. Q: If the man and his wife go on the recommended package tour, how much should they pay? A. $1,088. B. $1,958. (correct) C. $2,176. |

We extract 50 error cases of RekNet based on ALBERT\_base on DREAM randomly, finding 36% of them are related to logical calculation (especially numerical calculation), as Table \[XVIII\] shows. On the contrary, the original DREAM, RACE and Cosmos QA only have 14%, 8% and 4% examples requiring logical calculation respectively. Though Reference Span in RekNet can reveal computing words like discount, half from passages and questions, it fails to calculate the correct result due to the lack of human numeral logical operation, which calls for more in-depth researches in MRC field.

### VII. Conclusion

To alleviate the challenge of knowledge role missing in multi-choice MRC, this work makes the first attempt to integrate explicit knowledge based on reference span extraction into MRC modeling, presenting Reference Knowledgeable Network (RekNet), which can quote relevant and credible explicit knowledge for multi-choice MRC tasks. We evaluate the proposed RekNet on three typical multi-choice MRC benchmarks: RACE, DREAM and Cosmos QA, and obtain consistent and significant performance improvements which pass the significance tests. In the future, we will expand other common reading strategies to RekNet.

### References

[1] K. M. Hermann, T. Kociský, E. Grefenstette, L. Espeholt, W. Kay, M. Suleyman, and P. Blunsom, “Teaching machines to read and comprehend,” in Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, 2015, pp. 1693–1701.

[2] H. Wang, D. Yu, K. Sun, J. Chen, D. Yu, D. McAllester, and D. Roth, “Evidence sentence extraction for machine reading comprehension,” in Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL), 2019, pp. 696–707.

[3] R. Baradaran, R. Ghiassi, and H. Amirkhani, “A survey on machine reading comprehension systems,” arXiv preprint arXiv:2001.01582, 2020.
