REM-Net: Recursive Erasure Memory Network for Commonsense Evidence Refinement

Yinya Huang[1], Meng Fang[2], Xunlin Zhan[1], Qingxing Cao[1], Xiaodan Liang[1*]
[1] Shenzhen Campus of Sun Yat-sen University
[2] Tencent AI Lab / Robotics X
yinya.huang@hotmail.com, mfang@tencent.com,
{zhanxlin, caoqx}@mail2.sysu.edu.cn, xdliang328@gmail.com

Abstract

When answering a question, people often draw upon their rich world knowledge in addition to the particular context. While recent works retrieve supporting facts/evidence from commonsense knowledge bases to supply additional information to each question, there is still ample opportunity to advance it on the quality of the evidence. It is crucial since the quality of the evidence is the key to answering commonsense questions, and even determines the upper bound on the QA systems' performance. In this paper, we propose a recursive erasure memory network (REM-Net) to cope with the quality improvement of evidence. To address this, REM-Net is equipped with a module to refine the evidence by recursively erasing the low-quality evidence that does not explain the question answering. Besides, instead of retrieving evidence from existing knowledge bases, REM-Net leverages a pre-trained generative model to generate candidate evidence customized for the question. We conduct experiments on two commonsense question answering datasets, WIQA and CosmosQA. The results demonstrate the performance of REM-Net and show that the refined evidence is explainable.

Introduction

Commonsense question answering (commonsense QA) is recently an attractive field in that it requires systems to understand the common sense information beyond words, which are normal to human beings but nontrivial for machines. There are plenty of datasets that are proposed for this purpose, for instance, CommonsenseQA (Talmor et al. 2019), CosmosQA (Huang et al. 2019), WIQA (Tandon et al. 2019). Different from traditional machine reading comprehension (MRC) tasks such as SQuAD (Rajpurkar et al. 2016) or NewsQA (Trischler et al. 2016) that the key information for answering the questions is directly given by the context paragraph, solving commonsense questions requires a more comprehensive understanding of both the context and the relevant common knowledge, and further reasoning out the hidden logic between them. There are varieties of knowledge bases that meet the need, including text corpora like Wikipedia, and large-scale knowledge graphs (Speer, Chin, and Havasi 2017; Mitchell et al. 2015; Sap et al. 2019).

*Corresponding author: Xiaodan Liang.

Recent popular solution resorts to external supporting facts from such knowledge bases as evidence, to enhance the question with commonsense knowledge or the logic of reasoning (Devlin et al. 2019; Liu et al. 2019; Lv et al. 2020; Lin et al. 2019; Xu et al. 2020). However, the quality of the supporting facts is not guaranteed, as some of them are weak in interpretability so that do not help the question answering. Specifically, current methods are mainly two-fold. The first group of methods (Devlin et al. 2019; Liu et al. 2019; Bosselut et al. 2019) pre-train language models on those external supporting facts (e.g., Wikipedia, ConceptNet) so that the models could remember some of the common knowledge, which is empirically proven by Tandon et al. (2019) and Trinh and Le (2018). The second group of methods (Lv et al. 2020; Lin et al. 2019; Cao, Fang, and Tao 2019) incorporates the question with knowledge subgraphs or paths that carry information such as relation among concepts or show multi-hop reasoning process. The structured information is typically encoded via graph models such as GCN (Kipf and Welling 2016), and after which merged with the question features. Generally, current methods all handle evidence by brute force, without further selection or refinement according to the interpretability of the supporting facts. But as the example shown in Figure 1, some of the supporting facts do not interpret the question, regardless that they are semantically related. Thus, there is need for models that will further our processing of the evidence.

In this paper, we introduce a new recursive erasure memory network (REM-Net) that further refines the candidate supporting fact set. The REM-Net consists of three main components: a query encoder, an evidence generator, and a novel recursive erasure memory (REM) module. Specifically, the query encoder is a pre-trained encoder that encodes the question. The evidence generator is a pre-trained generative model that produces candidate supporting facts based on the question. Compared with those retrieved supporting facts, the generated facts provides new question-specific information beyond the existing knowledge bases. The REM module refines the candidate supporting fact set by recursively matching the supporting facts and the question in feature space to estimate each fact's quality. This estimation helps both updating the question feature and the supporting fact set. The question feature is updated by a residual term, whereas the supporting fact set is updated by remov-

Context
The seed germinates. The plant grows.
The plant flowers. Produces fruit.
The fruit releases seeds. The plant dies.
Question
Suppose less nutrients in the soil happens,
how will it affect less seeds germinates?
Answer Options
(A) More. (B) Less. (C) No effect.

Supporting Facts

✗ not is a good idea	is located at plant
✗ not made of iron	is created by plant
✓ causes starvation	is inherited from plant
is part of ecosystem	✓ is related to soil decay
is a symbol of decay	✗ is part of flower
has a less oxygen	is a plant
✓ ends with die	✓ requires soil
✓ not capable of grow	has a no life
✓ desires of water	desires of water
…	…

Figure 1: (a) An example about supporting facts for a question. The data is from WIQA (Tandon et al. 2019) dev set. The supporting facts are generated by COMET (Bosselut et al. 2019). The quality of the facts is not guaranteed. The facts are mostly semantically related to the key phrases in the question, but they contribute differently to answering this commonsense question. For example, "*is part of flower*" conveys an attribute of the concept "*seeds*", but does not tell us how in fact it will affect "*less seeds germinates*". By contrast, "*causes starvation*" gives straightforward information that fills the causal gap between "*less nutrients in the soil*" and "*less seeds germinates*". Therefore, facts like "*seeds is part of flower*" do not explain "*the cause of seeds germination*" or "*the effect of nutrients in the soil to the seeds germination*" that answers the question, whereas "*causes starvation*" as an evidence is favorable. (b) The facts with X marks are erased by our proposed REM-Net model, whereas those with check marks survive the multi-hop refinement.

ing the low-quality facts. Compared with the standard attention mechanisms (Xu et al. 2015; Vaswani et al. 2017) that allocate weights to the supporting facts once, the multi-hop operation in REM module widens the gap of how much each supporting fact contributes to the question answering by the number of recursive steps their features are incorporated for the feature update. Therefore this procedure leads to a refined use of given supporting facts.

We conduct experiments on two commonsense QA benchmarks, WIQA (Tandon et al. 2019) and CosmosQA (Huang et al. 2019). The experimental results demonstrate that REM-Net outperforms current methods, and the refined supporting facts are more qualified for the questions. Our contributions are mainly three-fold:

- We propose a model named recursive erasure memory network (REM-Net) towards evidence refinement according to the commonsense question, which improves the explainability of the supporting facts.
- We design a new REM module that recursively erases the unqualified supporting facts to provide refined appropriate evidence.
- Our experimental results demonstrate the superiority of REM-Net compared with other methods that uses external evidence. Moreover, case study shows the interpretability of the refined evidence.

Related Works

Commonsense Question Answering Similar to open-domain question answering tasks (Rajpurkar, Jia, and Liang 2018; Kwiatkowski et al. 2019), commonsense question answering (Tandon et al. 2019; Huang et al. 2019) requires open-domain information to support the answer prediction. But different from open-domain question answering tasks that the text comprehension is straightforward and the retrieved open-domain information is direct to the questions, in commonsense question answering tasks the open-domain information is more complicated in that they play a role as evidence to bridge the understanding gap in the commonsense questions. Current works leverage the open-domain information by whether incorporating external knowledge as evidence or training the models to generate evidence. Lv et al. (2020) extracts knowledge from ConceptNet (Speer, Chin, and Havasi 2017) and Wikipedia, and learns features with GCN (Kipf and Welling 2016) and graph attention (Veličković et al. 2017). Zhong et al. (2019) retrieves ConceptNet (Speer, Chin, and Havasi 2017) triplets and train two functions to measure direct and indirect connections between concepts. Rajani et al. (2019) train a GPT (Zhong et al. 2019) to generate reasonable evidence for the questions. During evaluation, the model generates evidence and predicts the multi-choice answers concurrently. Ye et al. (2019) automatically constructs a commonsense multi-choice dataset from ConceptNet triplets. However, the retrieved or generated evidence are usually not further refined, and some of them could be unnecessary or even confounding to answering the questions. The proposed model explores to refine the original evidence to discover those most supporting evidence to the commonsense questions and therefore provides stronger interpretations.

Memory Networks Memory networks (Weston, Chopra, and Bordes 2015; Bordes et al. 2015; Miller et al. 2016; Sukhbaatar et al. 2015) are proposed to solve early reasoning problems such as bAbI (Weston et al. 2016)) that requires to locate useful information for answer prediction. The sentences are stored into memory slots and later selected for the question answering. Recently, multi-head attention memory networks (Dai et al. 2019) are proposed so that takes advantage of the transformer-based networks. Our proposed model is based on multi-head attention memory network that is modified with a recursive erasure manipulation to adapt to the commonsense question answering tasks for accurate evidence refinement.

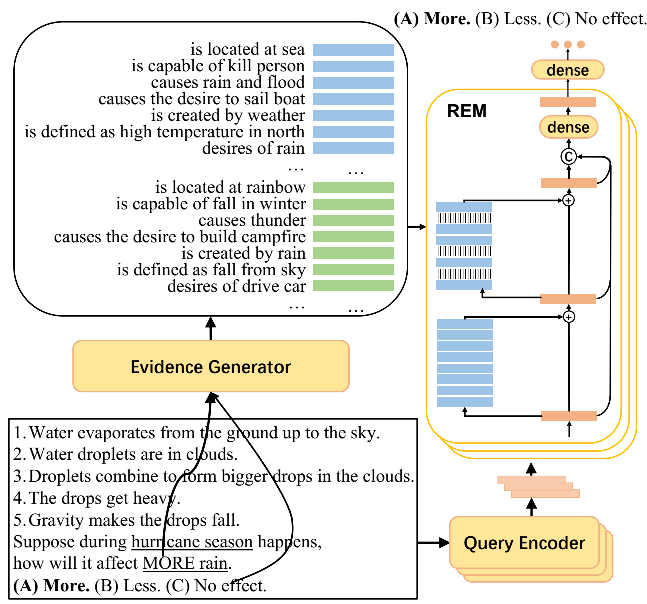

Figure 2: The proposed REM-Net with three main components: a query encoder that encodes the commonsense question; an evidence generator providing candidate evidence set in a generation manner; a recursive erasure memory (REM) module that conducts the evidence refinement.

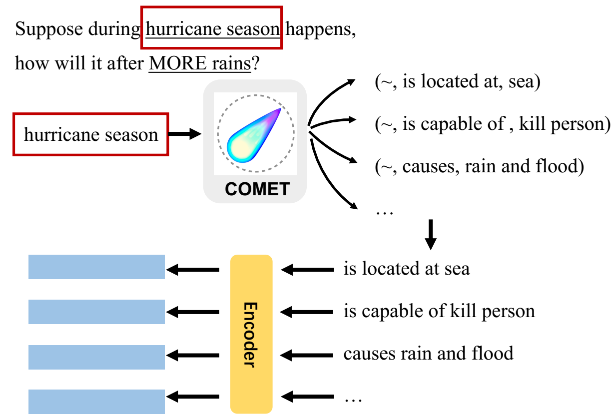

Figure 3: Details in the evidence generator. Key phrases are first extracted from the question with rules, then taken as triplet heads to generate relations and triplet tails by COMET (Bosselut et al. 2019). The triplets are turned into sentences, and finally encoded into evidence embeddings with a pre-trained encoder.

Recursive Erasure Memory Network

The main purpose of this model is to refine supporting facts so that they are more explainable to the question. The idea is to recursively erase the unqualified supporting facts. As a result, during the recursive procedure, the retained supporting facts are repeatedly used for updating the features.

The architecture of our model is shown in Figure 2. It has three main modules. A query encoder encodes the question to a query embedding. An evidence generator produces candidate supporting fact set, and encodes them into embeddings. A recursive erasure memory (REM) module refines the parameterized supporting facts by filtering out unqualified items conditioning on the query embedding.

Query Encoder

We follow baselines to use pre-trained language models (e.g., BERT (Devlin et al. 2019), RoBERTa (Liu et al. 2019)) to encode the question to contextual embeddings. Given a question as a triplet of (context paragraph, question sentence, answer options), the input sequence is in such format "`[CLS] context [SEP] question [SEP] answer option`", where "`[CLS]`" and "`[SEP]`" are special tokens for pre-trained language model. The output `[CLS]` embeddings are provided as query to the recursive erasure memory (REM) module.

Evidence Generator

Generally, for a commonsense question, its supporting facts can be obtained in three main sources: (1) retrieved texts/triplets from knowledge bases, (2) texts/triplets that are generated conditioning on the question, (3) reuse of the context paragraph. Among the three approaches, retrieval-based methods are widely used (Lv et al. 2020; Lin et al. 2019), whereas generation-based methods are barely explored. However, generated supporting facts provide new information that is beyond the commonsense question and knowledge bases. Therefore in this work we use generated supporting facts. We also compare the three sources of supporting facts in the experiment section.

The mechanism of the evidence generator are presented in Figure 3. The generation is achieved by four steps. First, it extracts key phrases from the question. Second, taking the key phrases as head concepts, it generates relations and tail concepts to complete ConceptNet-like triplets. This is implemented with COMET (Bosselut et al. 2019), a pre-trained model that is capable of generating commonsense knowledge triplets. Since the generation is based on the key phrases extracted from the question, the generated knowledge triplets are closely related to the question, but the combination of relations and concepts can be new to the existing knowledge bases. Third, the triplets are then converted into natural sentences according to COMET templates[1]. Finally, the sentences are encoded into embeddings with a pre-trained encoder.

Recursive Erasure Memory Module

The recursive erasure memory (REM) module takes the query embedding and the evidence matrix as input, producing an output feature that merges the updated embeddings. The detailed mechanism is shown in Figure 4. Similar to end-to-end memory networks (Sukhbaatar et al. 2015), REM module matches the question embedding and the evidence matrix recursively to find significant information for the question. However, the manipulations are essentially dif-

[1]https://mosaickg.apps.allenai.org/

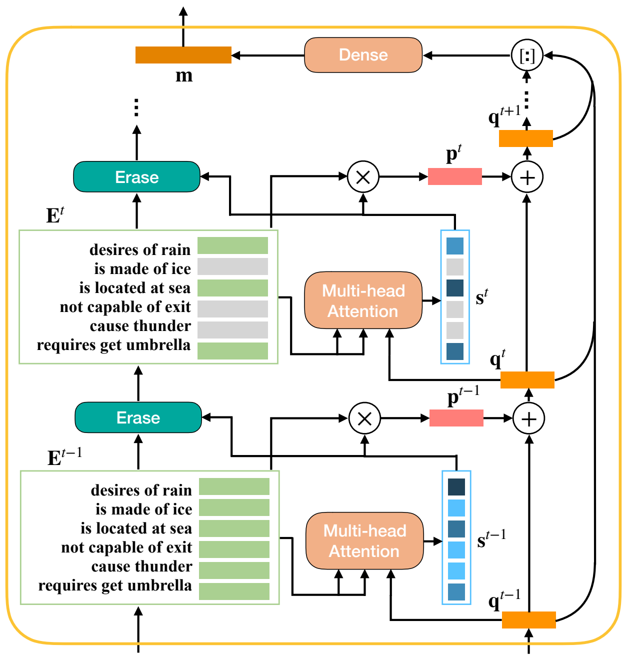

Figure 4: The recursive erasure memory (REM) module. (a) At each recursive step: (i) Multi-head attention (Vaswani et al. 2017) is applied to the evidence score estimation conditioning on the query embedding; (ii) A residual term is calculated by a cross product between the evidence matrix and the query embedding, after which is used for updating the query embedding; (iii) An erasure manipulation is conducted on the evidence set, where the supporting facts with low evidence score are filter from the evidence set. (b) At the end of the overall recursive procedure, the query embeddings at each recursive step are merged by concatenation and a linear projection.

ferent. Instead of taking inner product twice on separate evidence matrix to update the context information, REM module uses a single evidence matrix as both input and output, and conducts multi-head attention for information matching. The attention weights are then taken for updating both the evidence matrix and the query embedding.

Given the initial query embedding $\mathbf{q}^0$ from the query encoder and the initial evidence matrix $\mathbf{E}^0$ from the evidence generator, the first recursive step starts with a multi-head attention (Vaswani et al. 2017) that matches both information, so that each supporting fact is allocated with a weight as its evidence score:

$$\mathbf{s}^0 = \text{MultiHead}(\mathbf{q}^0, \mathbf{E}^0, \mathbf{E}^0). \tag{1}$$

The scores $\mathbf{s}^0$ are then used for updating both the query embedding $\mathbf{q}^0$ and the evidence matrix $\mathbf{E}^0$. The query embedding is updated with a residual term that is the outer product of the evidence score vector and the evidence matrix. Meanwhile, the evidence matrix is updated by erasing the low-ranked supporting facts that are sorted by the scores. The updated query embedding and the updated evidence matrix are then fed into the next recursive step. This procedure is recursively conducted until termination.

We formalize the manipulation at recursive step $t-1$. Current updated query embedding $\mathbf{q}^{t-1} \in \mathbb{R}^h$ and updated evidence matrix $\mathbf{E}^{t-1} \in \mathbb{R}^{I\times h}$ are fed into multi-head attention, where h is the embedding size and I is the number of stored supporting facts. $\mathbf{E}^{t-1}$ performs as the key and value and $\mathbf{q}^{t-1}$ as the query. We obtain evidence scores $\mathbf{s}^{t-1} \in \mathbb{R}^I$ for each supporting fact:

$$\mathbf{s}^{t-1} = \text{MultiHead}(\mathbf{q}^{t-1}, \mathbf{E}^{t-1}, \mathbf{E}^{t-1}). \tag{2}$$

The query embedding is updated with a residual term $\mathbf{p}^{t-1}$. It is the outer product of the evidence matrix $\mathbf{E}^{t-1}$ and the evidence score $\mathbf{s}^{t-1}$:

$$\begin{aligned} \mathbf{p}^{t-1} &= {\mathbf{E}^{t-1}}^\top \mathbf{s}^{t-1}, \\ \mathbf{q}^t &= \mathbf{q}^{t-1} + \mathbf{p}^{t-1}. \end{aligned} \tag{3}$$

The evidence matrix $\mathbf{E}^{t-1}$ is then updated with an erasure manipulation. According to the evidence scores, the supporting facts are sorted, and embeddings of the lowest k supporting facts are removed from the matrix. The evidence matrix is then updated to $\mathbf{E}^t$:

$$\mathbf{E}^t = \begin{bmatrix} \mathbf{e}_0^t \\ \mathbf{e}_1^t \\ \vdots \\ \mathbf{e}_I^t \end{bmatrix}, \ \mathbf{e}_i^t = \begin{cases} \mathbf{e}_i^{t-1}, & s_i^{t-1} \geq s_{[I-k]}^{t-1}, \\ \mathbf{0}, & s_i^{t-1} < s_{[I-k]}^{t-1}, \end{cases} \tag{4}$$

where $s_{[I-k]}^{t-1}$ is the lowest kth score among $\mathbf{s}^{t-1}$.

The resulting query $\mathbf{q}^t$ and evidence $\mathbf{E}^t$ are the inputs of the next recursive step. Therefore, the survived supporting facts are continually matched with the question, whereas the erased supporting facts stop contributing to this procedure. As a consequence, this multi-hop erasure manipulation provides more accurate and interpretable reasoning to the question answering, as the supporting facts are gradually refined.

At the end of the recursive procedure, queries in all recursive steps $\mathbf{q}^t, t \in \{0, 1, ..., T\}$ are concatenated and fed into a fully connected layer, as the output of the REM module:

$$\mathbf{m} = [\mathbf{q}^0; ...; \mathbf{q}^T]\mathbf{W}_m + \mathbf{b}_m, \tag{5}$$

where $[;]$ indicates the concatenation operation, $\mathbf{m} \in \mathbb{R}^h$, $\mathbf{W}_m \in \mathbb{R}^{hT\times h}$, and $\mathbf{b}_m \in \mathbb{R}^h$.

Answer Prediction

The probabilities Pr of choosing the final answer option are:

$$Pr = \text{SoftMax}([\mathbf{m}_1; ...; \mathbf{m}_C]\mathbf{W}_p + b_p), \tag{6}$$

where $[;]$ indicates concatenation, $\{\mathbf{m}_1, ..., \mathbf{m}_C\}$ are outputs of the REM module for each answer option, and C is the number of answer options. $\mathbf{W}_p \in \mathbb{R}^{h\times 1}$, $b_p \in \mathbb{R}$.

Experiments

We evaluate REM-Net on two commonsense QA datasets, WIQA (Tandon et al. 2019) and CosmosQA (Huang et al. 2019). We then conduct ablation study on the REM module, and show several cases of REM-Net's evidence refinement.

Method	In	Out	No	Total
Baselines				
Majority (2019)*	45.46	49.47	0.55	30.66
Polarity (2019)*	76.31	53.59	0.27	39.43
Adaboost (1995)*	49.41	36.61	48.42	43.93
Decomp-Attn (2016)*	56.31	48.56	73.42	59.48
Implicit use of evidence				
$\text{BERT}_{\text{BASE}}$(no para)	66.60	64.29	74.90	69.13
$\text{BERT}_{\text{BASE}}$	70.57	58.54	91.08	74.26
$\text{BERT}_{\text{BASE}}$(ensemble)	71.51	61.82	90.72	75.61
$\text{BERT}_{\text{LARGE}}$	73.40	63.88	90.52	76.69
$\text{BERT}_{\text{LARGE}}$(ensemble)	71.51	62.73	90.04	75.69
$\text{RoBERTa}_{\text{LARGE}}$	78.87	73.48	88.69	80.79
$\text{RoBERTa}_{\text{LARGE}}$(ensemble)	77.46	71.39	90.48	80.44
Explicit use of evidence				
MemN2N (2015)	38.50	38.01	39.52	38.85
Input Aug ($\text{BERT}_{\text{BASE}}$)	70.57	61.00	90.72	75.12
Input Aug ($\text{BERT}_{\text{LARGE}}$)	73.40	63.88	90.52	76.69
Input Aug ($\text{RoBERTa}_{\text{LARGE}}$)	75.66	71.59	90.60	80.25
SDP Att ($\text{BERT}_{\text{BASE}}$) (2017)	72.83	63.71	63.71	75.26
SDP Att ($\text{BERT}_{\text{LARGE}}$)	72.26	66.26	90.28	77.36
Ours				
REM-Net ($\text{BERT}_{\text{BASE}}$)	70.94	63.22	91.24	76.29
REM-Net ($\text{BERT}_{\text{LARGE}}$)	73.21	68.14	90.84	78.52
REM-Net ($\text{RoBERTa}_{\text{LARGE}}$)	76.23	69.13	92.35	80.09

Table 1: Results (accuracy%) on the WIQA test set, including accuracies on three separate question types (In="in-para", Out="out-of-para", No="no-effect"), and the overall test set. The baselines labeled with * are taken from Tandon et al. (2019), in which the used test set is slightly different.

Data

WIQA (Tandon et al. 2019) contains counterfactual questions in such a fixed pattern as "*suppose ... happens, how will it affects ...*", in which the two clauses relate to cause and effect separately. The context paragraphs provide descriptions of natural phenomenons, which are manually written based on specifically defined "influence graphs". The questions are split into three types ("in-para", "out-of-para", "no-effect") depending on whether the questions are derived from the original "influence graphs". For "out-of-para" and "no-effect" questions, the context paragraphs are irrelevant to the questions, so that they are unable to provide meaningful evidence.

CosmosQA (Huang et al. 2019) includes questions of daily life scenarios, such as cultural norms, counterfactual reasoning, situational fact, and temporal event. The scenarios are plentiful and the questions are also diverse. The questions are in a multi-choice format.

Compared Methods

We compare the performance of REM-Net with several groups of competitive methods.

Group 1: Baselines. For WIQA, Majority predicts the most frequent answer option in the training set. Polarity predicts answers with the most comparative words. Adaboost (Freund and Schapire 1995) uses bag-of-words features in the questions. Decomp-Attn (Parikh et al. 2016) is a decomposable attention model that computes attention between sentences. For CosmosQA, Sliding Window (Richardson, Burges, and Renshaw 2013) evaluates the similarity between context paragraph and answer options. Stanford Attentive Reader (Chen, Bolton, and Manning 2016), Gated-Attention Reader (Dhingra et al. 2017) and Co-Matching (Wang et al. 2018b) are reading comprehension systems that performs attention mechanism differently.

Method	Dev
Baselines	
Sliding Window (2013)	25.0
Stanford Attentive Reader (2016)	45.3
Gated-Attention Reader (2017)	46.9
Co-Matching (2018b)	45.9
Implicit use of evidence	
Commonsense-Rc (2018a)	47.6
GPT-FT (2018)	54.0
DMCN (2020)	67.1
$\text{BERT}_{\text{LARGE}}$(2019)	66.2
$\text{BERT}_{\text{LARGE}}$(ensemble)	67.1
$\text{BERT}_{\text{LARGE}}$(multiway) (2019)	68.3
$\text{RoBERTa}_{\text{LARGE}}$(2019)	78.6
Explicit use of evidence	
MemN2N (2015)	30.6
Input Aug ($\text{BERT}_{\text{LARGE}}$)	67.1
Input Aug ($\text{RoBERTa}_{\text{LARGE}}$)	80.8
SDP Att ($\text{BERT}_{\text{LARGE}}$)	27.4
SDP Att ($\text{RoBERTa}_{\text{LARGE}}$)	25.6
Ours	
REM-Net ($\text{BERT}_{\text{LARGE}}$)	69.5
REM-Net ($\text{RoBERTa}_{\text{LARGE}}$)	81.4

Table 2: Results (accuracy%) on the CosmosQA dev set.

Group 2: Implicit incorporation of supporting evidence. Commonsense-RC (Wang et al. 2018a) is an LSTM-based model pre-trained on RACE (Lai et al. 2017). Transformer-based pre-trained language models such as BERT (Devlin et al. 2019) and RoBERTa (Liu et al. 2019) that learn from large scale corpora.

Group 3: Explicit use of supporting evidence. End-to-end memory networks (MemN2N) (Sukhbaatar et al. 2015) are LSTM-based recursive models that recursively match the context to the question. Input augmentation (Input Aug) directly augments the question by appending the supporting evidence to the question text and encodes them to contextual embeddings with pre-trained language models. Scaled dot-product attention (SDP Att) (Vaswani et al. 2017) allocates attention weights to each supporting facts. In our experiments, the evidence is the same as REM-Net, which are supporting facts generated by the evidence generator.

	In	Out	No	Total
REM-Net (BERT$_{\text{LARGE}}$)	73.21	68.14	90.84	78.52
w/o E	69.81	52.79	91.79	72.09
w/o E, w/o R	62.08	43.27	92.35	67.10

Table 3: Ablation study on REM-Net (BERT$_{\text{LARGE}}$) that are conducted on WIQA. E signifies the erasure manipulation, whereas R indicates to the recursive mechanism. In="In-para" type, Out="Out-of-para" type, No="No-effect" type.

	Dev	Test
REM-Net (BERT$_{\text{LARGE}}$)	69.49	70.07
w/o E	68.44	68.58
w/o E, w/o R	68.27	68.53

Table 4: Ablation study on REM-Net (BERT$_{\text{LARGE}}$) that are conducted on CosmosQA. E denotes the erasure manipulation, while R refers to the recursive mechanism.

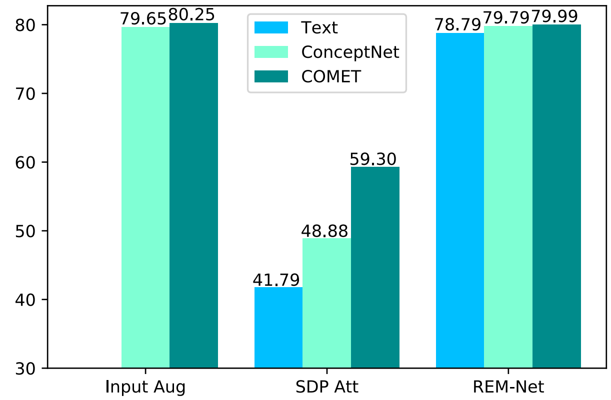

Figure 5: Comparison accuracies (%) on WIQA test set among three evidence sources. The base model being used for the three methods is RoBERTa$_{\text{LARGE}}$.

Experimental Setup

Seed Key Phrases Extraction The supporting facts are generated based on the key phrases. For WIQA, we set a rule to extract those key phrases. Since each of the question sentences consists of a cause clause and an effect clause with fixed pattern, we remove the pattern words to obtain two groups of key phrases, and separately generate two groups of supporting facts. For CosmosQA, we use the TAGME (Assante et al. 2019) toolkit [2] to automatically tag the key phrases from the context paragraphs and the question.

Implementation Details We use BERT (Devlin et al. 2019) and RoBERTa (Liu et al. 2019) as the backbones. The sequence length for the query encoder is 128, which is sufficient to include the input sequence "`[CLS] context [SEP] question [SEP] answer option`" ($> 88\%$). For the evidence generator, the sequence length is set to 30 and covers the vast majority of evidence sentences ($> 99\%$).

For experiments on WIQA, since there are two groups of supporting facts relating to the cause and the effect, we adopt two parallel REM modules to separately refine them. The model is optimized by Adam (Kingma and Ba 2015) with a learning rate of 1×10^{-5}. Warmup steps are 1000. We train 25 epochs with batch size 8. For the termination condition of the recursion, we set a fixed recursive step to 2. The upper bound of erased evidence sentences at each recursive step is 50. For CosmosQA, we use a single REM module to refine the evidence. The model is optimized using the Adam optimizer with a learning rate of 5×10^{-6} and warmup steps of 1500. The model is trained with 10 epochs and a batch size of 4. The fixed recursive step is 2. The upper bound of erased evidence sentences at each recursive step is 10.

Experimental Results

The experimental results are presented in Table 1 and Table 2. The REM-Net is compared with three groups of methods. It is shown that the REM-Net outperforms the compared approaches in most of the experiments. Besides, models perform differently on different data. In the CosmosQA dataset, our REM-Net outperforms all of the compared methods. In WIQA, REM-Net (BERT$_{\text{LARGE}}$) is superior, whereas REM-Net (RoBERTa$_{\text{LARGE}}$) is comparable to other methods. REM-Net (RoBERTa$_{\text{LARGE}}$) is mainly inferior in the "in-para" and "out-of-para" data type, but surpasses compared methods in the "no-effect" data type. This is because the majority of the "in-para" and "out-of-para" evidence is meaningful to the question, and thus the erasure operation from the REM module provides limited effect.

Ablation Study

We further investigate the details in REM-Net. The results are shown in Table 3 and Table 4. It is observed that removing the erasure manipulation from the REM module leads to performance drop. This indicates that excluding those low-quality supporting facts benefits the results. Further removing the recursive mechanism, which means the REM module calculates the evidence scores once, brings a further performance drop. This indicates that recursively estimating the evidence sentences refines the understanding of the question and provides better interpretation. Therefore, erasure manipulation and the recursive mechanism both contribute to the benefits provided by our model.

Generated Evidence Versus Retrieved Evidence

We compare the quality of generated evidence and retrieved evidence. For a fair comparison, both evidence are based on ConceptNet. Specifically, the generated evidence are produced by COMET that is pre-trained on ConceptNet, whereas the retrieved evidence is directly retrieved from ConceptNet. Besides, to provide baseline results, we simply take the context paragraph provided by the question as another type of evidence. In the experiments, we provide different types of evidence to three methods that use evidence in an explicit manner, which are input augmentation, scaled dot-product attention, and the proposed REM-Net. The comparison results are shown in Figure 5. It is shown that in the

[2] https://tagme.d4science.org/tagme/

Context The oil needs to be pumped from the ground. After it is pumped it then is transported to a factory. In the factory the oil is processed and turned into fuel. Once the fuel is refined it is then sent to a truck. By truck the fuel is sent to the gas station.	**Context** After 15 years of paying premiums to Allstate , I have finally started the process of shopping for a new insurance company . I ca n't say I ' ve been unhappy with Allstate but it 's time to see if they are truly giving me a good deal or not . A couple things have caused me to do this .	**Context** I was walking home from the store , when I saw an old man laying on the sidewalk , bleeding . The right side of his face was all covered in blood . He was conscious but seemed dazed and probably intoxicated . Nearby there was a young man dialing his cell phone .
Question and Options Suppose more oil is processed happens, how will it affect MORE oil arriving at gas stations ? (A) More. (B) Less. (C) No effect.	**Question and Options** Why is it a good idea to shop for insurance regularly ? (A) Sometimes your current insurance will be too complacent with you . (B) None of the above choices . (C) You need to keep your insurance provider on their toes. (D) It helps make sure that you are getting the best deal possible .	**Question and Options** What may happen after the young man makes his call ? (A) None of the above choices . (B) The bus would arrive at the stop soon . (C) The taxi would pick up the young man . (D) Medical personnel would come to help the old man .
Supporting facts ✗ The oil needs to be pumped from the ground. ✗ After it is pumped it then is transported to a factory. ✗ In the factory the oil is processed and turned into fuel. ✓ Once the fuel is refined it is then sent to a truck. ✓ By truck the fuel is sent to the gas station.	**Supporting facts** ✗ As a result, he/she feels sad. ✗ As a result, he/she feels good. ✗ As a result, he/she feels annoyed. ✗ As a result, he/she feels satisfied. ✗ As a result, he/she feels happy. ✓ Before, he/she needed have the information. ✓ Because he/she wanted to have good quality of products. ✓ He/she is seen as cautious. ✓ He/she is seen as smart. ✓ He/she is seen as responsible.	**Supporting facts** ✗ As a result, he/she wants put the phone down. ✗ As a result, he/she wants get a bandage. ✗ As a result, he/she wants to call a cab. ✗ As a result, he/she feels bad. ✗ Has an effect on he/she becomes scared. ✓ As a result, he/she wants go to jail. ✓ As a result, he/she wants call the police. ✓ Before, he/she needed pick up the phone. ✓ Because he/she wanted get money. ✓ He/she is seen careless.
(1) successful case (WIQA)	(2) successful case (CosmosQA)	(3) failure case (CosmosQA)

Figure 6: Examples of evidence refinement by REM-Net. Case (1) presents a successful case from the WIQA test set. The supporting facts are context sentences. Case (2) is a successful case from the CosmosQA dev set, in which the presented supporting facts are part of the generated facts by the evidence generator. Case (3) shows a failure case from the CosmosQA dev set. The presented supporting facts are part of the generated facts by the evidence generator, therein the underlined facts are incorrectly erased or retained.

three methods, incorporating the generated evidence gives better results than the retrieved evidence. The performance gap is especially obvious for scaled dot-product attention, since it selects the evidence once with the attention weights. The REM-Net refines the evidence in a multi-hop manner, and the performance gap between different evidence are small, but generated evidence still gives better result.

Case Study

We show three cases to see the qualify of refined evidence, as presented in Figure 6.

Figure 6 (1) shows a successful case in WIQA. The supporting facts are context paragraph sentences. The provided context paragraph covers a whole process of fuel production, whereas the question is about the causal relation between oil processing and fuel transportation. REM-Net erases the irrelevant oil processing sentences, retaining the sentences about fuel transportation.

Figure 6 (2) presents a successful case in CosmosQA, in which REM-Net refines generated supporting facts. The question is about good reasons for regularly buying insurance. The context paragraph tells a story about the narrator deciding to change his/her insurance products, but the reason for his/her decision is not provided. The generated facts supply such reasons. The erased facts such as "*As a result, he/she feels sad*" or "*As a result, he/she feels happy*" do not interprets the question, since changing the insurance products are normally someone's rational decision. On the contrary, "*Because he/she wanted to have good quality of products*" support the question well. It is intuitive that the retained facts interprets the question better.

Figure 6 (3) shows a failure case in CosmosQA. This question is about the follow-up events after the young man makes a call to help the old man. The erasure by REM-Net seems unreasonable. The erased supporting facts include "*As a result, he/she wants put the phone down*" and "*As a result, he/she wants get a bandage*", which are events related to the question. On the other hand, the retained supporting facts contain "As a result, he/she wants go to fail" and "Because he/she wanted get money". Including the context and the question, these supporting facts are unreasonable inferences. This case indicates that the erasure operation of REM-Net does not cover all the questions. One of the reasons is that the commonsense questions are in varied domains so that some of the domains with fewer samples are not well trained.

Conclusion

In this paper, we propose a recursive erasure memory network (REM-Net) that refines evidence for commonsense question. It recursively estimates quality of each supporting fact based on the question, and refines the supporting fact set accordingly. The recursive procedure leads to repeated use of high-quality supporting facts, so that the question answering is conducted by useful information. Experimental results demonstrates that REM-Net is effective for the commonsense QA tasks, and the evidence refinement is interpretable. Besides, we evaluate the quality of generated evidence compared to retrieved evidence, learning that using generated evidence gives better performance.

Acknowledgments

The authors would like to thank Yu Cao, Jinghui Qin, Zheng Ye, Zhicheng Yang for their useful discussions. This work was supported in part by National Natural Science Foundation of China (NSFC) under Grant No.U19A2073 and No.61976233, Guangdong Province Basic and Applied Basic Research (Regional Joint Fund-Key) Grant No.2019B1515120039, Shenzhen Basic Research Project (Project No. JCYJ20190807154211365), Zhijiang Lab's Open Fund (No. 2020AA3AB14) and CSIG Young Fellow Support Fund.

References

Assante, M.; Candela, L.; Castelli, D.; Cirillo, R.; Coro, G.; Frosini, L.; Lelii, L.; Mangiacrapa, F.; Pagano, P.; Panichi, G.; et al. 2019. Enacting open science by D4Science. *Future Generation Computer Systems* 101: 555–563.

Bordes, A.; Usunier, N.; Chopra, S.; and Weston, J. 2015. Large-scale Simple Question Answering with Memory Networks. *ArXiv, abs/1506.02075* .

Bosselut, A.; Rashkin, H.; Sap, M.; Malaviya, C.; Celikyilmaz, A.; and Choi, Y. 2019. COMET: Commonsense Transformers for Automatic Knowledge Graph Construction. In *Proc. of ACL*.

Cao, Y.; Fang, M.; and Tao, D. 2019. Bag: Bi-directional attention entity graph convolutional network for multi-hop reasoning question answering. In *Proc. of NAACL*.

Chen, D.; Bolton, J.; and Manning, C. D. 2016. A Thorough Examination of the CNN/Daily Mail Reading Comprehension Task. In *Proc. of ACL*.

Dai, Z.; Dai, W.; Liu, Z.; Rao, F.; Chen, H.; Zhang, G.; Ding, Y.; and Liu, J. 2019. Multi-Task Multi-Head Attention Memory Network for Fine-Grained Sentiment Analysis. In *Proc. of NLPCC*.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proc. of NAACL*.

Dhingra, B.; Liu, H.; Yang, Z.; Cohen, W. W.; and Salakhutdinov, R. 2017. Gated-Attention Readers for Text Comprehension. In *Proc. of ACL*.

Freund, Y.; and Schapire, R. E. 1995. A desicion-theoretic generalization of on-line learning and an application to boosting. In *European conference on computational learning theory*, 23–37. Springer.

Huang, L.; Bras, R. L.; Bhagavatula, C.; and Choi, Y. 2019. Cosmos QA: Machine Reading Comprehension with Contextual Commonsense Reasoning. In *Proc. of EMNLP*.

Kingma, D. P.; and Ba, J. L. 2015. Adam: A Method for Stochastic Optimization. In *Proc. of ICLR*.

Kipf, T. N.; and Welling, M. 2016. Semi-Supervised Classification with Graph Convolutional Networks. *ArXiv, abs/1609.02907* .

Kwiatkowski, T.; Palomaki, J.; Redfield, O.; Collins, M.; Parikh, A. P.; Alberti, C.; Epstein, D.; Polosukhin, I.; Devlin, J.; Lee, K.; Toutanova, K.; Jones, L.; Kelcey, M.; Chang, M.-W.; Dai, A. M.; Uszkoreit, J.; Le, Q.; and Petrov, S. 2019. Natural Questions: A Benchmark for Question Answering Research. *Proc. of ACL* .

Lai, G.; Xie, Q.; Liu, H.; Yang, Y.; and Hovy, E. H. 2017. RACE: Large-scale ReAding Comprehension Dataset From Examinations. In *Proc. of EMNLP*.

Lin, B. Y.; Chen, X.; Chen, J.; and Ren, X. 2019. KagNet: Knowledge-Aware Graph Networks for Commonsense Reasoning. In *Proc. of EMNLP*.

Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; and Stoyanov, V. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *ArXiv, abs/1907.11692* .

Lv, S.; Guo, D.; Xu, J.; Tang, D.; Duan, N.; Gong, M.; Shou, L.; Jiang, D.; Cao, G.; and Hu, S. 2020. Graph-Based Reasoning over Heterogeneous External Knowledge for Commonsense Question Answering. *Proc. of AAAI* .

Miller, A. H.; Fisch, A.; Dodge, J.; Karimi, A.-H.; Bordes, A.; and Weston, J. 2016. Key-Value Memory Networks for Directly Reading Documents. In *Proc. of EMNLP*.

Mitchell, T.; Cohen, W.; Hruschka, E.; Talukdar, P.; Betteridge, J.; Carlson, A.; Dalvi, B.; Gardner, M.; Kisiel, B.; Krishnamurthy, J.; Lao, N.; Mazaitis, K.; Mohamed, T.; Nakashole, N.; Platanios, E.; Ritter, A.; Samadi, M.; Settles, B.; Wang, R.; Wijaya, D.; Gupta, A.; Chen, X.; Saparov, A.; Greaves, M.; and Welling, J. 2015. Never-Ending Learning. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence (AAAI-15)*.

Parikh, A. P.; Tackstrom, O.; Das, D.; and Uszkoreit, J. 2016. A Decomposable Attention Model for Natural Language Inference. In *Proc. of EMNLP*.

Radford, A.; Narasimhan, K.; Salimans, T.; and Sutskever, I. 2018. Improving language understanding with unsupervised learning. *Technical report, OpenAI* .

Rajani, N. F.; McCann, B.; Xiong, C.; and Socher, R. 2019. Explain Yourself! Leveraging Language Models for Commonsense Reasoning. In *Proc. of ACL*.

Rajpurkar, P.; Jia, R.; and Liang, P. 2018. Know What You Don't Know: Unanswerable Questions for SQuAD. *Proc. of ACL* .

Rajpurkar, P.; Zhang, J.; Lopyrev, K.; and Liang, P. 2016. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250* .

Richardson, M.; Burges, C. J.; and Renshaw, E. 2013. MCTest: A Challenge Dataset for the Open-Domain Machine Comprehension of Text. In *Proc. of EMNLP*.

Sap, M.; Bras, R. L.; Allaway, E.; Rashkin, H.; Bhagavatula, C.; Lourie, N.; Roof, B.; Smith, N.; and Choi, Y. 2019. ATOMIC: An Atlas of Machine Commonsense for If-Then Reasoning. *Proc. of AAAI* .

Speer, R.; Chin, J.; and Havasi, C. 2017. ConceptNet 5.5: An Open Multilingual Graph of General Knowledge. In *Proc. of AAAI*.

Sukhbaatar, S.; Szlam, A.; Weston, J.; and Fergus, R. 2015. End-To-End Memory Networks. *Proc. of NIPS* .

Talmor, A.; Herzig, J.; Lourie, N.; and Berant, J. 2019. CommonsenseQA: A Question Answering Challenge Targeting Commonsense Knowledge. In *NAACL-HLT 2019: Annual Conference of the North American Chapter of the Association for Computational Linguistics*, 4149–4158.

Tandon, N.; Dalvi, B.; Sakaguchi, K.; Clark, P.; and Bosselut, A. 2019. WIQA: A dataset for "What if..." reasoning over procedural text. In *Proc. of EMNLP*.

Trinh, T. H.; and Le, Q. V. 2018. A Simple Method for Commonsense Reasoning. *ArXiv, abs/1806.02847* .

Trischler, A.; Wang, T.; Yuan, X.; Harris, J.; Sordoni, A.; Bachman, P.; and Suleman, K. 2016. Newsqa: A machine comprehension dataset. *arXiv preprint arXiv:1611.09830* .

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Łukasz Kaiser; and Polosukhin, I. 2017. Attention is all you need. In *Proc. of NIPS*.

Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; and Bengio, Y. 2017. Graph Attention Networks. *Proc. of ICLR* .

Wang, L.; Sun, M.; Zhao, W.; Shen, K.; and Liu, J. 2018a. Yuanfudao at SemEval-2018 Task 11: Three-way Attention and Relational Knowledge for Commonsense Machine Comprehension. In *Proc. of SemEval*.

Wang, S.; Yu, M.; Jiang, J.; and Chang, S. 2018b. A Co-Matching Model for Multi-choice Reading Comprehension. In *Proc. of ACL*.

Weston, J.; Bordes, A.; Chopra, S.; Rush, A. M.; van Merriënboer, B.; Joulin, A.; and Mikolov, T. 2016. Towards AI-Complete Question Answering: A Set of Prerequisite Toy Tasks. In *Proc. of ICLR*.

Weston, J.; Chopra, S.; and Bordes, A. 2015. Memory Networks. In *Proc. of ICLR*.

Xu, K.; Ba, J.; Kiros, R.; Cho, K.; Courville, A.; Salakhudinov, R.; Zemel, R.; and Bengio, Y. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *Proc. of ICML*, 2048–2057.

Xu, Y.; Fang, M.; Chen, L.; Du, Y.; Zhou, J. T.; and Zhang., C. 2020. Deep Reinforcement Learning with Stacked Hierarchical Attention for Text-based Games. In *Proc. of NeurIPS*.

Ye, Z.-X.; Chen, Q.; Wang, W.; and Ling, Z.-H. 2019. Align, Mask and Select: A Simple Method for Incorporating Commonsense Knowledge into Language Representation Models. *ArXiv, abs/1908.06725* .

Zhang, S.; Zhao, H.; Wu, Y.; Zhang, Z.; Zhou, X.; and Zhou, X. 2020. DCMN+: Dual Co-Matching Network for Multi-choice Reading Comprehension. *Proc. of AAAI* .

Zhong, W.; Tang, D.; Duan, N.; Zhou, M.; Wang, J.; and Yin, J. 2019. Improving Question Answering by Commonsense-Based Pre-training. In *Proc. of NLPCC*.