Commonsense Evidence Generation and Injection in Reading Comprehension

Ye Liu[1], **Tao Yang**[2], **Zeyu You**[2], **Wei Fan**[2] **and Philip S. Yu**[1]
[1]Department of Computer Science, University of Illinois at Chicago, IL, USA
[2]Tencent Hippocrates Research Lab, Palo Alto, CA, USA
{yliu279, psyu}@uic.edu, {tytaoyang, davidwfan}@tencent.com, youz@onid.orst.edu

Abstract

Human tackle reading comprehension not only based on the given context itself but often rely on the commonsense beyond. To empower the machine with commonsense reasoning, in this paper, we propose a **C**ommonsense **E**vidence **G**eneration and **I**njection framework in reading comprehension, named **CEGI**. The framework injects two kinds of auxiliary commonsense evidence into comprehensive reading to equip the machine with the ability of rational thinking. Specifically, we build two evidence generators: one aims to generate textual evidence via a language model; the other aims to extract factual evidence (automatically aligned text-triples) from a commonsense knowledge graph after graph completion. Those evidences incorporate contextual commonsense and serve as the additional inputs to the reasoning model. Thereafter, we propose a deep contextual encoder to extract semantic relationships among the paragraph, question, option, and evidence. Finally, we employ a capsule network to extract different linguistic units (word and phrase) from the relations, and dynamically predict the optimal option based on the extracted units. Experiments on the CosmosQA dataset demonstrate that the proposed CEGI model outperforms the current state-of-the-art approaches and achieves the highest accuracy (83.6%) on the leaderboard.

1 Introduction

Contextual commonsense reasoning has long been considered as the core of understanding narratives (Hobbs et al., 1993; Andersen, 1973) in reading comprehension (Charniak and Shimony, 1990). Despite the broad recognition of its importance, the research of reasoning in narrative text is limited due to the difficulty of understanding the causes and effects within the context. Comprehending reasoning requires not only understanding the explicit meaning of each sentence but also making inferences based on implicit connections between sentences.

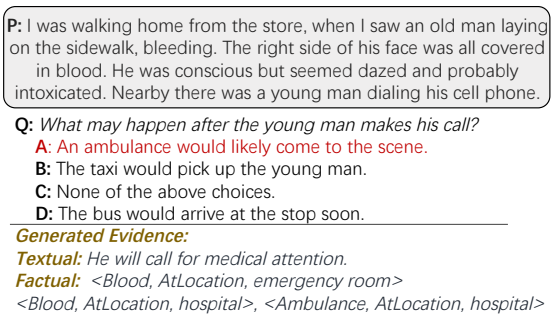

Figure 1: Example of generated evidence helping answer the commonsense question.

To answer a contextual commonsense question correctly, two important characteristics need to be well considered. First, the information that is required to infer a correct answer may be beyond the context, and hence adding external commonsense knowledge to guide the reasoning is necessary. Second, how to use external knowledge to gain contextual understanding between the paragraph, question and option set is difficult but important. Despite the great success of large pre-trained models such as BERT (Devlin et al., 2018), GPT (Radford et al., 2018) and RoBERTa (Liu et al., 2019), recent studies suggest that those models fail to capture sufficient knowledge and provide commonsense inference. For example, Poerner et al. (2019) show that language models perform well in reasoning about entity names, but fail to capture rich factual knowledge. Moreover, Talmor et al. (2019) state that language models fail on half of the reasoning tasks which require symbolic operations such as comparison, conjuction and composition.

To this end, we introduce a **C**ommonsense **E**vidence **G**eneration and **I**njection framework in reading comprehension, named **CEGI**, which generates useful evidence from textual and factual knowledge and injects the generated evidence into pre-trained models such as RoBERTa. We propose

Proceedings of the SIGdial 2020 Conference, pages 61–73
1st virtual meeting, 01-03 July 2020.

to generate evidence regarding the facts and their relations. More specifically, we use language models to generate textual evidence and extract factual evidence from a knowledge graph after graph completion. We then inject both evidences into the proposed contextual commonsense reasoning model to predict the optimal answer. As shown in Figure 1, the *Textual Generated Evidence* "He will call for medical attention" and *Factual Generated Evidence* "both blood & ambulance locate at hospital" can help the model find the correct answer "An ambulance would likely come to the scene".

To capture relations between the paragraph and question, many reading comprehension models (Zhang et al., 2019a; Tang et al., 2019) have been proposed. However, those reasoning models are essentially based on the given context without understanding the facts behind. Moreover, in many situations, the candidate option set contains distractors that are quite similar to the correct answer. In other words, understanding the relations among the option set is also important. We employ a capsule network (Sabour et al., 2017), which uses a routing-by-agreement mechanism to capture the correlations among different options and make the final decision.

Our proposed CEGI framework not only utilizes external commonsense knowledge to generate reasoning evidence but also adopts a capsule network to make the final answer prediction. The explainable evidence and the ablation studies indicate that our method has a large impact on the performance of the commonsense reasoning in reading comprehension. The contributions of this paper are summarized as follows: 1) We introduce two evidence generators which are learned from textual and factual knowledge sources; 2) We provide an injection method that can infuse both evidences into the contextual reasoning model; 3) We adapt a capsule network to our reasoning model to capture interactions among candidate options when making a decision; 4) We show our **CEGI** model outperforms current state-of-the-art models on the CosmosQA dataset and generates richer interpretive evidence which helps the commonsense reasoning.

2 Related Work

2.1 Multi-choice Reading Comprehension

To model the relation and alignment between the pairs of paragraph, question and option set, various approaches seek to use attention and pursue deep representation for prediction. Tang et al. (2019) and Wang et al. (2018b) model the semantic relationships among paragraph, question and candidate options from multiple aspects of matching. Zhu et al. (2018a) propose a hierarchical attention flow model, which leverages candidate options to capture the interactions among paragraph, question and candidate options. Chen et al. (2019) merge various attentions to fully extract the mutual information among the paragraph, question and options and form the enriched representations.

2.2 Commonsense Knowledge Injection

To empower the model with human commonsense reasoning, various approaches have been proposed on the context-free commonsense reasoning task. The majority of the approaches are focusing on finding the question entity and a reasoning path on the knowledge graph to obtain the answer entity (Huang et al., 2019; Zellers et al., 2018; Talmor et al., 2018). For an instance, Lin et al. (2019) construct graphs to represent relevant commonsense knowledge, and then calculate the plausibility score of the path between the question and answer entity. Lv et al. (2019) extract evidence from both structured knowledge base and unstructured texts to build a relational graph and utilize graph attention to aggregate graph representations to make final predictions. However for contextual commonsense reasoning, it's hard to find a single most relevant entity from the paragraph or question to obtain the correct answer.

Other approaches focus on enhancing the pre-trained language models through injecting external knowledge into the model and updating the model parameters in multi-task learning (Zhang et al., 2019b; Lauscher et al., 2019; Levine et al., 2019). A knowledge graph injected ERNIE model is introduced in (Zhang et al., 2019b) and a weakly supervised knowledge-pretrained language model (WkLM) is introduced in (Xiong et al., 2019). They both inject the knowledge through aligning the source with the fact triplets in WikiData. However, the parameters need to be retrained when injecting new knowledge, which could lead to the catastrophic forgetting (McCloskey and Cohen, 1989).

3 Task Definition

In multi-choice reading comprehension, we are given a paragraph $\mathbf{P}$ with t tokens $\mathbf{P} = [\mathrm{p}_1, \mathrm{p}_2, \ldots, \mathrm{p}_\mathrm{t}]$, a question $\mathbf{Q}$ containing n tokens

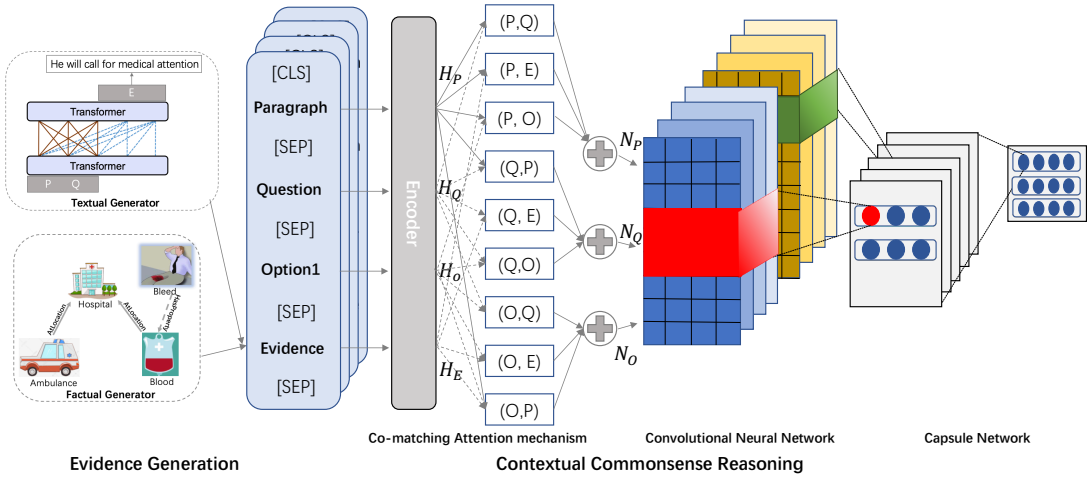

Figure 2: The proposed commonsense evidence generation and injection (**CEGI**) framework.

$\mathbf{Q} = [\mathrm{q}_1, \mathrm{q}_2, \ldots, \mathrm{q}_\mathrm{n}]$ and the option set with m candidate options $\mathbf{O} = \{\mathbf{O}_1, \mathbf{O}_2, \ldots, \mathbf{O}_\mathrm{m}\}$, where each candidate option is a text with h tokens $\mathbf{O}_\mathrm{i} = [\mathrm{o}_1, \mathrm{o}_2, \ldots, \mathrm{o}_\mathrm{h}]$. The goal is to select the correct answer $\mathbf{A}$ from the candidate option set. For simplicity, we denote $\mathcal{X} = \{\mathbf{P}, \mathbf{Q}, \mathbf{O}\}$ as one data sample and denote $\mathbf{y} = [\mathrm{y}_1, \mathrm{y}_2, \ldots, \mathrm{y}_\mathrm{m}]$ as a one-hot label, where each scale $\mathrm{y}_\mathrm{i} = \mathbf{1}(\mathbf{O}_\mathrm{i} = \mathbf{A})$ is an indicator function. In the training stage, we are given N set of $(\mathcal{X}, \mathbf{y})^N$, the goal is to learn a model $f : \mathcal{X} \rightarrow \mathbf{y}$. In the testing, we need to predict $\mathbf{y}^{\mathrm{test}}$ given test samples $\mathcal{X}^{\mathrm{test}}$.

When answering a question according to the paragraph, we observe that the context itself often does not provide enough clues to guide us to the correct answer. To this end, we need to know comprehensive information beyond the context and perform commonsense reasoning. Hence, we split the task into two parts: evidence generation and answer prediction, respectively. Our proposed **CEGI** model addresses both parts accordingly by two generators: textual evidence generator and factual evidence generator. In textual evidence generator, our goal is to generate relevant evidence text $\mathbf{E} = [\mathrm{e}_1, \mathrm{e}_2, \ldots, \mathrm{e}_\mathrm{k}]$ given question $\mathbf{Q}$ and paragraph $\mathbf{P}$. Note that the number of evidence tokens k may vary in different question and paragraph pair. In factual evidence generator, the goal is to generate relevant text that describes the relations between facts where the facts are the entities from paragraph, question and options. In the second part, we aim to learn a classifier $P(\mathbf{y}|\mathbf{P}, \mathbf{Q}, \mathbf{O}, \mathbf{E})$ that predicts the correct option when a new data sample is given. By using the evidence generated from the first part, we expect the reasoning model can be enhanced with the auxiliary information, especially for those questions that require contextual commonsense reasoning.

4 Methodology

To tackle reading comprehension task with commonsense reasoning, we introduce a commonsense evidence generation and injection (**CEGI**) framework. The system diagram of the CEGI framework is shown in Fig. 2. First, the evidence generation module produces textual evidence and factual evidence. Those generated evidences will be used as auxiliary inputs for the reasoning model. Second, the contextual commonsense reasoning module generates deep contextual features for the paragraph, question, option and evidence. Meanwhile, a bidirectional attention mechanism is applied to the features to capture representations of the pair of paragraph, question, option set and evidence. Next, all pairs are concatenated and fed into a convolutional neural network for extracting different linguistic units of the options. At least, a capsule network is then applied to dynamically update the representation vector of the candidate options. The final answer is one of the options with the largest vector norm. We describe more details of each component in the following subsections.

4.1 Evidence Generation

It is worthy to mention that many commonsense reasoning types, such as causes of events and effects of events, are important factors of understanding the context in reading comprehension. While those factors are often not explicit or given in the paragraph and option set, answering such may be-

come difficult. To this end, we seek to learn relevant evidence that contains commonsense knowledge. Specifically, we leverage pretrained language models to learn from both context and knowledge graph that may contain reasoning relations. We exploit two kinds of generators, textual evidence generator and factual evidence generator.

4.1.1 Textual Evidence Generator

We observe that daily life events often follow a common routine such that when one event happened, the resulting event or the cause of such an event follows a specific pattern. For an example, in Figure 1, the given paragraph describes a scenario that the old man is hurt and the young man is making a phone call. If we know that he is calling for medical attention, answering the question would become easy. Hence, the goal of our proposed textual evidence generator is to generate the text that follows daily life event routines. We rely on a pretrained language model to acquire the textual evidence by using GPT2 (Radford et al., 2018) and Uniml (Dong et al., 2019). Specifically, in the training, we concatenate the paragraph, question and the correct answer as the input to the standard language model (Liu et al., 2018). Accordingly, the textual evidence generated from the language model is the following sentence after the question text. Note that we stack [**P** [SEP] **Q** [SEP] **A**] as the input to train the language model. Formally, let $[\mathrm{w}^1, \ldots, \mathrm{w}^\mathrm{T}] = [\mathbf{P}\ [\mathrm{SEP}]\ \mathbf{Q}\ [\mathrm{SEP}]\ \mathbf{A}]$. The language generation model aims to maximize the following likelihood (Radford et al., 2018):

$$\mathcal{L}_{gen} = \sum\nolimits_{\mathrm{i}=1}^{T} p(\mathrm{w}^\mathrm{i}|\mathrm{w}^1, ..., \mathrm{w}^{\mathrm{i}-1}), \qquad (1)$$

where the conditional probability $p(\mathrm{w}^\mathrm{i}|\mathrm{w}^1, \ldots, w^{\mathrm{i}-1}) = f(\mathrm{w}^1, \ldots, \mathrm{w}^{\mathrm{i}-1})$ and f is a sequence of operations that (i) converts each token w^i into token embedding $\mathbf{W}_\mathrm{e}^\mathrm{i}$ and position embedding $\mathbf{W}_\mathrm{p}^\mathrm{i}$; (ii) transforms them into features with L layers where each layer feature is $\mathbf{H}^\mathrm{l}(\mathrm{w}^\mathrm{i}) = \mathrm{h}^\mathrm{l}(\mathrm{g}(\mathbf{W}_\mathrm{e}^{\mathrm{i}-1}, \mathbf{W}_\mathrm{p}^{\mathrm{i}-1}), \mathbf{H}^\mathrm{l}(\mathrm{w}^{\mathrm{i}-1}))$, and (iii) converts the feature into a probability using a linear classifier by predicting the next token w^i.

Moreover, we aim to generate evidence that can discriminate the correct answer from option distractors. Hence, we add the answer prediction loss into the objective to fine-tune the language model. The text input for the jth option is $\mathbf{x}_\mathrm{j} = [\mathbf{P}\ [\mathrm{SEP}]\ \mathbf{Q}\ [\mathrm{SEP}]\ \mathbf{O}_\mathrm{j}]$. We use all N samples to optimize the following objective (with a regularization term λ):

$$\mathcal{L}_{class} = \sum_{(x,y)\in\{\mathcal{X},\mathbf{y}\}} \log(\mathrm{Softmax}(\mathbf{H}^\mathrm{L}(\mathrm{w}^0)\mathrm{W}_\mathrm{y})), \qquad (2)$$

$$\mathcal{L}_{total} = \mathcal{L}_{gen} + \lambda * \mathcal{L}_{class}, \qquad (3)$$

where $\mathbf{H}^L(w^0)$ is the last layer feature of the first token and W_y is the parameters to learn to predict label y.

Test stage: we only use [**P** [SEP] **Q**] as the input to the language model and use the model to generate the next sentence as an evidence which means model is agnostic to the correct answer.

4.1.2 Factual Evidence Generator

Aside from the textual evidence that contains information about the facts of daily life routine, relations between the facts are also important for question answering. In this section, we propose to utilize a factual knowledge graph to extract facts and relations and use them as additional evidence. Specifically, we use the ConceptNet (Speer et al., 2017)[1] as the base model. We use a knowledge graph completion algorithm Bosselut et al. (2019) to find new relations to further improve the quality of the generated factual evidence.

We define $X^s = \{x_0^s, ..., x_{|s|}^s\}$ as the subject, $X^r = \{x_0^r, ..., x_{|r|}^r\}$ as the relation, and $X^o = \{x_0^o, ..., x_{|o|}^o\}$ as the object. We use the $[X^s\ [\mathrm{SEP}]\ X^r\ [\mathrm{SEP}]\ X^o]$ triplets as the input to the knowledge graph completion language model in Bosselut et al. (2019) to generate additional triplets that contain new subject and object relations. To generate factual evidence, we first extract entities from the given data $\mathcal{X}$. We then select the related entities that match the subject X^s in forms of subject-relation-object triplets. After that, we filter the triplets by selecting the subject X^{s*} that follows: (i) part-of-speech (POS) tag of X^{s*} word matches the POS tag of the entity word; (ii) subject X^{s*} word frequency is less than the word frequency of the object X^o plus a threshold K^o; (iii) subject X^{s*} word is not in the top-K frequent words based on the word frequency table[2]; and (iv) the relation X^r in the (X^{s*}, X^r, X^o) triplets connects no more than K^r objects from the same subject X^{s*}. K, K^o and K^r are the hyper-parameters. Finally, we convert the

[1] ConceptNet is a knowledge graph, which consists of triples obtained from the Open Mind Common Sense entries.

[2] https://www.wordfrequency.info/free.asp

filtered triplets into a nature language sequences as our factual evidences. For example, "(trouble, Partof, life)" would be converted to "trouble is part of life".

4.2 Model Learning with Contextual Commonsense Reasoning

After the relevant reasoning evidences are generated, the goal is to combine the evidence with the given data and then build a reasoning model to make a selection for the correct answer. In the following, we introduce our proposed contextual commonsense reasoning module, which utilizes contextual encoding, evidence injection and a capsule network components to make the prediction.

Contextual Encoding Recently, RoBERTa (Liu et al., 2019) has shown to be effective and powerful in many natural language processing tasks and it is potentially beneficial for generating deep contextual features as well. Here, we use RoBERTa as an intermediate component to generate hidden representation of paragraph, question, the ith option and evidence $[\mathbf{H}^{\mathrm{i}}_{\mathrm{cls}}, \mathbf{H}^{\mathrm{i}}_{\mathrm{P}}, \mathbf{H}^{\mathrm{i}}_{\mathrm{sep}} \mathbf{H}^{\mathrm{i}}_{\mathrm{Q}}, \mathbf{H}^{\mathrm{i}}_{\mathrm{sep}} \mathbf{H}^{\mathrm{i}}_{\mathrm{O_i}}, \mathbf{H}^{\mathrm{i}}_{\mathrm{sep}}, \mathbf{H}^{\mathrm{i}}_{\mathrm{E}}] = \mathrm{Encode}([[\mathrm{CLS}], \mathbf{P}, [\mathrm{SEP}], \mathbf{Q}, [\mathrm{SEP}], \mathbf{O}_{\mathrm{i}}, [\mathrm{SEP}], \mathbf{E}])$. We use the last layer of the RoBERTa model to encode, and thus the function $\mathrm{Encode}(\cdot)$ returns the last layer features for each token. The corresponding features of paragraph, question, option and evidence are $\mathbf{H}^{\mathrm{i}}_{\mathrm{P}} \in \mathcal{R}^{\mathrm{d}\times\mathrm{t}}$, $\mathbf{H}^{\mathrm{i}}_{\mathrm{Q}} \in \mathcal{R}^{\mathrm{d}\times\mathrm{n}}$, $\mathbf{H}^{\mathrm{i}}_{\mathrm{O_i}} \in \mathcal{R}^{\mathrm{d}\times\mathrm{h}}$ and $\mathbf{H}^{\mathrm{i}}_{\mathrm{E}} \in \mathcal{R}^{\mathrm{d}\times\mathrm{k}}$, where d is the dimension of the feature. Since we have m options, we have m set of features.

Evidence Injection Given the previously generated evidence representation $\mathbf{H}^{\mathrm{i}}_{\mathrm{E}}$. We aim to integrate it with the paragraph $\mathbf{H}^{\mathrm{i}}_{\mathrm{P}}$, question $\mathbf{H}^{\mathrm{i}}_{\mathrm{Q}}$ and option $\mathbf{H}^{\mathrm{i}}_{\mathrm{O_i}}$. Here, we adopt the attention mechanism used in QANet (Yu et al., 2018) to model the interaction between $\mathbf{H}^{\mathrm{i}}_{\mathrm{E}}$ and the paragraph $\mathbf{H}^{\mathrm{i}}_{\mathrm{P}}$:

$$\mathbf{S}^{\mathrm{E}}_{\mathrm{iP}} = \mathrm{Att}(\mathbf{H}^{\mathrm{i}}_{\mathrm{E}}, \mathbf{H}^{\mathrm{i}}_{\mathrm{P}}) = \mathrm{Softmax}(\mathbf{H}^{\mathrm{iT}}_{\mathrm{P}} \mathrm{W_g} \mathbf{H}^{\mathrm{i}}_{\mathrm{E}}) \tag{4}$$

$$\mathbf{G}^{\mathrm{E}}_{\mathrm{iP}} = \mathbf{H}^{\mathrm{i}}_{\mathrm{E}} \mathbf{S}^{\mathrm{E^T}}_{\mathrm{iP}}, \tag{5}$$

where $\mathrm{W_g} \in \mathcal{R}^{\mathrm{d}\times\mathrm{d}}$ is the bi-linear model parameter matrix. Since $\mathbf{S}^{\mathrm{E}}_{\mathrm{iP}} \in \mathcal{R}^{\mathrm{t}\times\mathrm{k}}$ is the activation map (attention weights) between each token in $\mathbf{P}$ and each token in $\mathbf{E}$, the learned relation representation $\mathbf{G}^{\mathrm{E}}_{\mathrm{iP}} \in \mathcal{R}^{\mathrm{d}\times\mathrm{t}}$ of the paragraph $\mathbf{P}$ contains evidence information $\mathbf{E}$. The other two relations $\mathbf{G}^{\mathrm{Q}}_{\mathrm{iP}}$ and $\mathbf{G}^{\mathrm{O_i}}_{\mathrm{iP}}$ regarding $\mathbf{P}$ can be generated accordingly. Similarly, we can model the other interactions for question $\mathbf{Q}$ as $\mathbf{G}^{\mathrm{P}}_{\mathrm{iQ}}, \mathbf{G}^{\mathrm{E}}_{\mathrm{iQ}}, \mathbf{G}^{\mathrm{O_i}}_{\mathrm{iQ}}$, and each option $\mathbf{O}_{\mathrm{i}}$ as $\mathbf{G}^{\mathrm{Q}}_{\mathrm{iO_i}}, \mathbf{G}^{\mathrm{E}}_{\mathrm{iO_i}}$ and $\mathbf{G}^{\mathrm{P}}_{\mathrm{iO_i}}$.

To incorporate the relation information, we use the co-matching algorithm introduced in Wang et al. (2018b) to generate the final representation of the input. First, we obtain the matching result between the paragraph and the question as follows:

$$\mathbf{M}^{\mathrm{Q}}_{\mathrm{iP}} = (\mathrm{W_m}[\mathbf{G}^{\mathrm{Q}}_{\mathrm{iP}} \ominus \mathbf{H}^{\mathrm{i}}_{\mathrm{P}}; \mathbf{G}^{\mathrm{Q}}_{\mathrm{iP}} \odot \mathbf{H}^{\mathrm{i}}_{\mathrm{P}}] + \mathrm{b_m} \otimes \mathbf{1})^{+}, \tag{6}$$

where $(\cdot)^{+}$ denotes ReLU function, $\mathbf{1} = [1, 1, \ldots, 1]^T \in \mathcal{R}^{t\times 1}$ is vector of all ones, and $\mathrm{W_m} \in \mathcal{R}^{\mathrm{d}\times 2\mathrm{d}}$ and $\mathrm{b_m} \in \mathcal{R}^{\mathrm{d}\times 1}$ are the model parameters. Following Tai et al. (2015) and Wang et al. (2018b), we use notation $\ominus$ and $\odot$ as the element-wise subtraction and multiplication between two matrices and $\otimes$ as outer product of two vectors. Similarly, we can obtain the other pairs as $\mathbf{M}^{\mathrm{E}}_{\mathrm{iP}}, \mathbf{M}^{\mathrm{O_i}}_{\mathrm{iP}}, \ldots, \mathbf{M}^{\mathrm{P}}_{\mathrm{iO_i}}$. In the next step, we concatenate all the pairs regarding $\mathbf{P}$ as

$$\mathbf{C}_{\mathrm{iP}} = \left[\mathbf{M}^{\mathrm{Q}}_{\mathrm{iP}} : \mathbf{M}^{\mathrm{O_i}}_{\mathrm{iP}} : \mathbf{M}^{\mathrm{E}}_{\mathrm{iP}}\right] \in \mathcal{R}^{3\mathrm{d}\times\mathrm{t}}, \tag{7}$$

where $[:]$ denotes the vertical concatenation operation. Each column $\mathbf{c}_{\mathrm{i}}$ is the co-matching state that concurrently matches a paragraph token with the question, candidate option and the evidence. Accordingly, we can obtain the question representation $\mathbf{C}_{\mathrm{iQ}}$ and option representation $\mathbf{C}_{\mathrm{iO_i}}$. Finally, we concatenate them all to obtain the final representation $\mathbf{F} = [\mathbf{C}_1, \ldots, \mathbf{C}_{\mathrm{m}}] \in \mathcal{R}^{3\mathrm{d}\times\mathrm{m}(\mathrm{t}+\mathrm{n}+\mathrm{h})}$, where each $\mathbf{C}_{\mathrm{i}} = [\mathbf{C}_{\mathrm{iP}}, \mathbf{C}_{\mathrm{iQ}}, \mathbf{C}_{\mathrm{iO_i}}] \in \mathcal{R}^{3\mathrm{d}\times(\mathrm{t}+\mathrm{n}+\mathrm{h})}$.

Since the final representation only contains the fine-grid token-level information, we employ a convolutional neural network (CNN) to extract higher level (phrase-level) patterns. To generate phrase patterns with different size, we use two convolutional kernels: size 1×2 with stride 2 and size 1×4 with stride 4 to convolve with $\mathbf{F}$ along the dimension of hidden state. In other words, such an operation extracts non-overlapping moving windows on $\mathbf{F}$ with window size 2 and 4.

$$\mathbf{R}_1 = \mathrm{MaxPooling}_{1\times 2}\{\mathrm{CNN}_{1\times 2}(\mathbf{F})\}$$
$$\mathbf{R}_2 = \mathrm{MaxPooling}_{1\times 1}\{\mathrm{CNN}_{1\times 4}(\mathbf{F})\}$$

To ensure $\mathbf{R}_1$ and $\mathbf{R}_2$ have the same dimension, we use a max pooling of size 1×2 with stride 2 for $\mathbf{R}_1$ and a max pooling of size 1×1 with stride 1 for $\mathbf{R}_2$. We concatenate $\mathbf{R}_1$ and $\mathbf{R}_2$ to generate phrase-level representation $\mathbf{L} = [\mathbf{R}_1, \mathbf{R}_2] \in \mathcal{R}^{3\mathrm{d}\times\mathrm{m}((\mathrm{t}+\mathrm{n}+\mathrm{h})/2)}$.

With $\mathbf{L}$, to predict the final answer, one of the commonly applied operation is to simply take the maximum over the hidden dimension of length $(\mathrm{t}+\mathrm{n}+\mathrm{h})/2$. However, the max operation only consider the most significant phrase for each candidate without aware of the others. To explore the correlation between options and dynamically select the optimal one, we use dynamic routing-by-agreement algorithm represented in Sabour et al. (2017). Specifically, we convert $\mathbf{L}_\mathrm{i}$ to a capsule $\mathbf{v}_\mathrm{j}$ using the following steps:

$$\hat{\mathbf{L}}_{\mathrm{j|i}} = W_{\mathrm{ij}}\mathbf{L}_\mathrm{i}, \quad \mathbf{s}_\mathrm{j} = \sum_{\mathrm{i}=1}^{(\mathrm{t}+\mathrm{n}+\mathrm{h})/2} \mathrm{c}_{\mathrm{ij}} \cdot \hat{\mathbf{L}}_{\mathrm{j|i}},$$
$$\mathbf{v}_\mathrm{j} = \frac{||\mathbf{s}_\mathrm{j}||^2}{1+||\mathbf{s}_\mathrm{j}||^2} \frac{\mathbf{s}_\mathrm{j}}{||\mathbf{s}_\mathrm{j}||},$$

where $\mathbf{L}_\mathrm{i}$ is the ith column vector of $\mathbf{L}$, affine transformation matrix W_{ij} and weighting c_{ij} are the capsule network model parameters. The learned $\hat{\mathbf{L}}_{\mathrm{j|i}}$ denotes the "vote" of the capsule j for the input capsule i. The agreement of "prediction vector" $\hat{\mathbf{L}}_{\mathrm{j|i}}$ between the current jth output and ith parent capsule is captured by the coupling coefficients c_{ij}. The value of c_{ij} would increase if higher level capsule $\mathbf{s}_\mathrm{j}$ and lower lever capsule $\mathbf{L}_\mathrm{i}$ highly agreed.

Model Learning If an option $\mathbf{O}_\mathrm{j}$ is the correct answer, we would like the top-level capsule $\mathbf{v}_\mathrm{j}$ to have a high energy, otherwise, we expect the energy of $\mathbf{v}_\mathrm{j}$ to be low. Since the L_2-norm (square root of the energy) of the capsule vector $\mathbf{v}_\mathrm{j}$ represents the scoring of how likely the jth candidate is the correct answer, we use the following loss function (Sabour et al., 2017) to learn the model parameters:

$$\mathcal{L}_{\mathrm{pre}} = \sum_{\mathrm{j}=1}^{\mathrm{m}} \{\mathrm{y_i} \cdot \max(0, \mathrm{m}^+ - ||\mathbf{v}_\mathrm{j}||)^2 + \lambda_1(1-\mathrm{y_i})\max(0, ||\mathbf{v}_\mathrm{j}|| - \mathrm{m}^-)^2\} \quad (8)$$

where λ_1 is a down-weighting coefficient, m^+ and m^- are margins. In our experiments, we set $m^+ = 0.9$, $m^- = 0.1$, $\lambda_1 = 0.5$.

5 Experiments

In the experiment, we evaluate the performance of our proposed CEGI framework from different aspects, including evidence generation tasks and the answer prediction of contextual commonsense reasoning tasks.

5.1 Dataset and Baseline

CosmosQA is the dataset that is designed for reading comprehension with commonsense reasoning (Huang et al., 2019). Samples are collected from people's daily narratives and the type of questions are concerning the causes or effects of events. Particularly answering the questions require contextual commonsense reasoning over the considerably complex, diverse, and long context. In general, the dataset contains a total of 35.2K multiple-choice questions, including 25262 training samples, 2985 development samples, and 6963 testing samples.[3]

Baseline We categorize baseline methods into the following three groups: 1. Co-Matching (Wang et al., 2018b), Commonsense-RC (Wang et al., 2018a), DMCN (Zhang et al., 2019a), Multiway (Huang et al., 2019). 2. GPT2-FT (Radford et al., 2018), BERT-FT (Devlin et al., 2018), RoBERTa-FT (Liu et al., 2019). 3. Commonsense-KB (Li et al., 2019), K-Adapter (Wang et al., 2020). The baseline details are in appendix A.2.

Table 1: Comparison of approaches on CosmosQA (Accuracy %) from the AI2 Leaderboard. T+F means using generated textual and factual evidence together.

Model	Dev	Test
Co-Matching (Wang et al., 2018b)	45.9	44.7
Commonsense-RC (Wang et al., 2018a)	47.6	48.2
DMCN (Zhang et al., 2019a)	67.1	67.6
Multiway (Huang et al., 2019)	68.3	68.4
GPT-FT (Radford et al., 2018)	54.0	54.4
BERT-FT (Devlin et al., 2018)	66.2	67.1
RoBERTa-FT (Liu et al., 2019)	79.4	79.2
Commonsense-KB (Li et al., 2019)	59.7	\
K-Adapter (Wang et al., 2020)	81.8	\
CEGI(T+F)	**83.8**	**83.6**
Human	\	94.0

5.2 Experimental Results and Analysis

Table 1 shows the performance of different approaches reported on the AI2 Leaderboard.[4]. Comparing to all methods, our proposed model CEGI(T+F) has the highest accuracy on both development set and test set. Most of the reading comprehension approaches utilize the attention mechanism to capture the correlations between paragraph, question and option set, therefore, the model tends to select the one option that is semantically closest to the paragraph. Among all of the group 1 methods, Multiway has the highest accuracy of 68.3%.

[3]The CosmosQA dataset can be obtained from https://leaderboard.allenai.org/cosmosqa/

[4]https://leaderboard.allenai.org/cosmosqa/ The test dataset is hidden by the AI2 and methods like Commonsense-KB and K-Adapter are not reported on the Leaderboard.

Group 2 methods consider deep contextual representation of the given paragraph, question and option set, and increase the performance. Comparing group 2 methods with group 1 methods, RoBERTa-FT, which uses dynamic masking and large mini-batches strategy to train BERT, gains 11.1% accuracy increase compared to Multiway.

However, it is worthy to mention that more than 83% of correct answers are not in the given passages in the CosmosQA dataset. Hence, multi-choice reading comprehension models do not gain big improvement as they tend to select the choice which has the most overlapped words with the paragraph without commonsense reasoning. Even though, group 2 methods consider connecting the paragraph with question and option through a deep bi-directional strategy, the reasoning for question answering is still not well-addressed in the models. By utilizing additional knowledge, CommonsenseKB or K-Adapter teach pretrained models with commonsense reasoning. K-Adapter gains 2.4% accuracy increase than RoBERTa-FT. Those approaches leverage the structured knowledge but fail to produce a prominent prediction improvement. Comparing our CEGI approach with RoBERTa, we gain a 4% increase and 2% increase than K-Adapter, which demonstrates that injecting evidence is beneficial and incorporating interactive attentions can further enhance the model.

5.3 Evidence Evaluation

In this section, we investigate the generated evidence from the textual generator and factual generator. Moreover, we study the quality of the generated evidence on another dataset—CommonsenseQA.

5.3.1 Textual Evidence Generator

Dataset Open Mind Common Sense (OMCS) corpus (Singh et al., 2002) is a crowd-sourced knowledge database of commonsense statements [5], where its English dataset contains a million sentences from over 15,000 contributors. We consider using this dataset to pretrain the textual evidence generator and using CosmosQA to fine-tune the generator.

Setup We use both BERT and GPT2 model to generate evidence and compare the results. To obtain a language model that contains representation of facts, we first pretrain both models with the OMCS data using the loss function in Eq. 1. Then we use CosmosQA data to fine-tune the pretrained model using multi-task loss in Eq. 3.

Metrics In line with prior work (Wang and Cho, 2019), we evaluate the performance of evidence generation based on quality and diversity. In terms of quality, we follow Yu et al. (2017) and compute the BLEU score between the generated evidence and the ground truth evidence to measure the similarity. The perplexity (PPL) score is also reported as a proxy for fluency. In terms diversity, we consider using self-BLEU (Zhu et al., 2018b), which measures how similar between two generated sentences. Generally, a higher self-BLEU score implies that the model has a lower diversity.

Results From Table 2, we observe that, compared to CEGI-GPT2, the CEGI-BERT generator has higher diversity (Self-BLEU decreases 4 for bi-gram and decreases 2.1 for tri-gram) but lower quality (BLEU decreases 1.3 for tri-gram and PPL increases 27.1). Even though the perplexity on CEGI-BERT is as good as CEGI-GPT2, after reading the samples, we find out that many of the generated language are fairly coherent. For a more rigorous measure of generation quality, we collect human judgments on sentences for 100 samples using a four-point scale (the higher the better). For each sample, we ask three annotators to rate the sentence on its fluency and take the average of the three judgments as the sentence's fluency score. For CEGI-BERT and CEGI-GPT2, we get mean scores of 3.21, 3.17 respectively. Those results imply that generated evidence are semantically consistent with the correct evidence and can be used as auxiliary knowledge for the reasoning step.

Table 2: Generation performance on CosmosQA.

Model	Quality			Diversity	
	BLEU(↑)		PPL(↓)	Self-BLEU(↓)	
	n=2	n=3		n=2	n=3
CEGI-BERT	**40.8**	32.2	153.8	**30.5**	**14.7**
CEGI-GPT2	39.8	**33.5**	**126.7**	34.2	16.6

Table 3: Generation performance on ConceptNet

Model	PPL	Score	N/T sro	N/T o
LSTM-s	\	60.83	**86.25**	7.83
CKBG	\	57.17	**86.25**	**8.67**
CEGI-BERT	4.89	92.19	65.32	4.12
CEGI-GPT2	**4.58**	**93.89**	61.72	3.90

[5] https://github.com/commonsense/conceptnet5/wiki/Downloads

5.3.2 Factual Evidence Generator

Dataset ConceptNet[6] is a commonsense knowledgebase of the most basic things a person knows. We use the 100K version of the training set in ConceptNet, which contains 34 relation types, to train the factual evidence generator. Tuples within the data are in the standard $< s, r, o >$ form.

Setup We set s and r as input for both GPT2 and BERT and use them to generate the new object o. To compare with our proposed GPT2 model and BERT model, we include a LSTM model (LSTM-s) and the BiLSTM model (CKBG) in (Saito et al., 2018). We train the LSTM model to generate o, and we train the CKBG model from both directions: s, r as input and o, r as input.

Metrics Similar to the textual evidence generation task, we use PPL to evaluate our model on relation generation. To evaluate the quality of generated knowledge, we also report the number of generated positive examples that are scored by the Bilinear AVG model (Li et al., 2016). "N/T sro" and "N/T o" are the proportions of generated tuples and generated objects which are not in the training set.

Results As we observed from Table 3, CEGI-GPT2 has the lowest PPL (4.58) and highest score (93.89), which indicates the CEGI-GPT2 model is confident and accurate at the generated relations. Even though the generated tuples on LSTM-s and CKGB model has high "N/T sro" (both are 86.25%) and "N/T o" (7.83% and 8.67% respectively), which means they generate novel relations and expand the knowledge graph, the generated nodes and relations may not be correct. We still need to rely on the Score to evaluate and they do poorly (60.83% and 57.17% respectively) in terms of Score. Since our proposed CEGI-GPT2 and CEGI-BERT model have high Score and low PPL, we believe that both models can produce high-quality knowledge and still be able to extend the size of the knowledge graph.

5.3.3 Evidence Evaluation on CommonsenseQA

CommonsenseQA[7] is a multi-choice question answering dataset, which contains roughly 12K questions with one correct answer and four distractor answers. Since the CommonsenseQA data only requires different types of commonsense knowledge to predict the correct answers, it does not contain paragraphs compared to CosmosQA. We use our textual generator and factual generator to generate evidence using CommonsenseQA data and use that to test the performance on answer prediction. To train our proposed textual evidence generator, we use Cos-e[8] as the ground truth evidence. Cos-e uses Amazon Mechanical Turk to provide reasoning explanations for the CommonsenseQA dataset. To train our proposed factual evidence generator, we follow the same procedure as described in subsection 4.1.2. To predict the answer based on both evidence, we prepare the input as [**Q** [SEP] $\mathbf{O}_i$ [SEP] **E**] to the RoBERTa model.

Baselines KagNet (Lin et al., 2019), Cos-E (Rajani et al., 2019), DREAM (Lv et al., 2019), RoBERTa + KE, RoBERTa + IR and RoBERTa + CSPT (Lv et al., 2019). All baselines use extracted knowledge from ConceptNet or Wikipedia. The details are in the appendix A.2.

Table 4: Accuracy (%) of different models on CommonsenseQA development set

Model	Acc
KagNet (Lin et al., 2019)	62.4
Cos-E (Rajani et al., 2019)	64.7
DREAM (Lv et al., 2019)	73.0
RoBERTa+CSPT (Lv et al., 2019)	76.2
RoBERTa+KE (Lv et al., 2019)	77.5
RoBERTa+IR (Lv et al., 2019)	78.9
RoBERTa + T	78.8
RoBERTa + F	77.6
RoBERTa + (T+F)	**79.1**

Result Results on CommonsenseQA datasets are summarized in Table 4. RoBERTa + T, RoBERTa + F and RoBERTa + (T+F) includes textual evidence, factual evidence and both evidence together respectively. We observe that our model RoBERTa + T and RoBERTa + F can produce competitive performance compared to all baselines. By utilizing both textual knowledge and factual knowledge, our approach outperforms RoBERTa+IR and achieves the highest accuracy 79.1%.

5.4 Ablation Study

To evaluate the contributions of individual components of our proposed framework, we use an ablation study. Table 5 summarizes ablation studies on the development set of CosmosQA from several aspects: the influence of the generated evidence; which evidence is better, textual or factual; the influence of the capsule network.

[6] https://ttic.uchicago.edu/~kgimpel/commonsense.html

[7] https://www.tau-nlp.org/commonsenseqa

[8] https://github.com/salesforce/cos-e

Result We can see that injecting generated explainable evidence can help the model achieve a better performance in terms of accuracy. Using generated textual evidence and factual evidence together can benefit more. Using capsule network significantly improves the reasoning performance, we doubt that is due to the hierarchical structure information from both token-level and phrase-level are extracted by capsule network.

Table 5: Accuracy (%) of different models on Cosmos development set. ✔ means selecting the module.

Model	Text	Fact	Capsule	Co-Att	Acc
CEGI	✔	✔	✔	✔	83.8
CEGI-V1	✔		✔	✔	83.4
CEGI-V2		✔	✔	✔	83.2
CEGI-V3			✔	✔	82.6
CEGI-V4	✔	✔			82.2
RoBERTa-FT					79.4

6 Conclusion

In this paper, we proposed a commonsense evidence generation and injection model to tackle reading comprehension. Both textual and factual evidence generators were used to enhance the model for answering questions which requires commonsense reasoning. After the evidences were generated, we adopted attention mechanism to find the relation and match between paragraph, question, option and evidence. We used convolutional network to capture the multi-grained features. To capture diverse features and iteratively make a decision, we proposed using a capsule network that dynamically capture different features to predict the answer. The AI2 Leaderboard of CosmosQA task demonstrated that our method can tackle commonsense-based reading comprehension pretty well and it outperformed the current state-of-the-art approach K-Adapter with a 2% increase in term of accuracy. Experiments regarding the evidence generators showed that the generated evidence is human-readable and those evidences are helpful for the reasoning task.

7 Acknowledge

This work is supported in part by NSF under grants III-1526499, III-1763325, III-1909323, and CNS-1930941.

References

Henning Andersen. 1973. Abductive and deductive change. *Language*, pages 765–793.

Antoine Bosselut, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, Asli Celikyilmaz, and Yejin Choi. 2019. Comet: Commonsense transformers for automatic knowledge graph construction. *arXiv preprint arXiv:1906.05317*.

Eugene Charniak and Solomon Eyal Shimony. 1990. *Probabilistic semantics for cost based abduction*. Brown University, Department of Computer Science.

Zhipeng Chen, Yiming Cui, Wentao Ma, Shijin Wang, and Guoping Hu. 2019. Convolutional spatial attention model for reading comprehension with multiple-choice questions. *Proceedings of the AAAI*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. Unified language model pre-training for natural language understanding and generation. *arXiv preprint arXiv:1905.03197*.

Jerry R Hobbs, Mark E Stickel, Douglas E Appelt, and Paul Martin. 1993. Interpretation as abduction. *Artificial intelligence*, 63(1-2):69–142.

Lifu Huang, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2019. Cosmos qa: Machine reading comprehension with contextual commonsense reasoning. *arXiv preprint arXiv:1909.00277*.

Anne Lauscher, Ivan Vulić, Edoardo Maria Ponti, Anna Korhonen, and Goran Glavaš. 2019. Informing unsupervised pretraining with external linguistic knowledge. *arXiv preprint arXiv:1909.02339*.

Yoav Levine, Barak Lenz, Or Dagan, Dan Padnos, Or Sharir, Shai Shalev-Shwartz, Amnon Shashua, and Yoav Shoham. 2019. Sensebert: Driving some sense into bert. *arXiv preprint arXiv:1908.05646*.

Shiyang Li, Jianshu Chen, and Dian Yu. 2019. Teaching pretrained models with commonsense reasoning: A preliminary kb-based approach. *arXiv preprint arXiv:1909.09743*.

Xiang Li, Aynaz Taheri, Lifu Tu, and Kevin Gimpel. 2016. Commonsense knowledge base completion. In *Proceedings of the 54th ACL (Volume 1: Long Papers)*, pages 1445–1455.

Bill Yuchen Lin, Xinyue Chen, Jamin Chen, and Xiang Ren. 2019. Kagnet: Knowledge-aware graph networks for commonsense reasoning. *arXiv preprint arXiv:1909.02151*.

Peter J Liu, Mohammad Saleh, Etienne Pot, Ben Goodrich, Ryan Sepassi, Lukasz Kaiser, and Noam Shazeer. 2018. Generating wikipedia by summarizing long sequences. *arXiv preprint arXiv:1801.10198*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Shangwen Lv, Daya Guo, Jingjing Xu, Duyu Tang, Nan Duan, Ming Gong, Linjun Shou, Daxin Jiang, Guihong Cao, and Songlin Hu. 2019. Graph-based reasoning over heterogeneous external knowledge for commonsense question answering. *arXiv preprint arXiv:1909.05311*.

Michael McCloskey and Neal J Cohen. 1989. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pages 109–165. Elsevier.

Nina Poerner, Ulli Waltinger, and Hinrich Schütze. 2019. Bert is not a knowledge base (yet): Factual knowledge vs. name-based reasoning in unsupervised qa. *arXiv preprint arXiv:1911.03681*.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding with unsupervised learning. Technical report, Technical report, OpenAI.

Nazneen Fatema Rajani, Bryan McCann, Caiming Xiong, and Richard Socher. 2019. Explain yourself! leveraging language models for commonsense reasoning. *arXiv preprint arXiv:1906.02361*.

Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. 2017. Dynamic routing between capsules. In *NeurIPS*, pages 3856–3866.

Itsumi Saito, Kyosuke Nishida, Hisako Asano, and Junji Tomita. 2018. Commonsense knowledge base completion and generation. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 141–150.

Push Singh, Thomas Lin, Erik T Mueller, Grace Lim, Travell Perkins, and Wan Li Zhu. 2002. Open mind common sense: Knowledge acquisition from the general public. In *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*, pages 1223–1237. Springer.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Proceddings of the AAAI*.

Kai Sheng Tai, Richard Socher, and Christopher D Manning. 2015. Improved semantic representations from tree-structured long short-term memory networks. *arXiv preprint arXiv:1503.00075*.

Alon Talmor, Yanai Elazar, Yoav Goldberg, and Jonathan Berant. 2019. olmpics–on what language model pre-training captures. *arXiv preprint arXiv:1912.13283*.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2018. Commonsenseqa: A question answering challenge targeting commonsense knowledge. *arXiv preprint arXiv:1811.00937*.

Min Tang, Jiaran Cai, and Hankz Hankui Zhuo. 2019. Multi-matching network for multiple choice reading comprehension. *Proceddings of the AAAI*.

Alex Wang and Kyunghyun Cho. 2019. Bert has a mouth, and it must speak: Bert as a markov random field language model. *arXiv preprint arXiv:1902.04094*.

Liang Wang, Meng Sun, Wei Zhao, Kewei Shen, and Jingming Liu. 2018a. Yuanfudao at semeval-2018 task 11: Three-way attention and relational knowledge for commonsense machine comprehension. *arXiv preprint arXiv:1803.00191*.

Ruize Wang, Duyu Tang, Nan Duan, Zhongyu Wei, Xuanjing Huang, Cuihong Cao, Daxin Jiang, Ming Zhou, et al. 2020. K-adapter: Infusing knowledge into pre-trained models with adapters. *arXiv preprint arXiv:2002.01808*.

Shuohang Wang, Mo Yu, Shiyu Chang, and Jing Jiang. 2018b. A co-matching model for multi-choice reading comprehension. *arXiv preprint arXiv:1806.04068*.

Wenhan Xiong, Jingfei Du, William Yang Wang, and Veselin Stoyanov. 2019. Pretrained encyclopedia: Weakly supervised knowledge-pretrained language model. *arXiv preprint arXiv:1912.09637*.

Adams Wei Yu, David Dohan, Minh-Thang Luong, Rui Zhao, Kai Chen, Mohammad Norouzi, and Quoc V Le. 2018. QANet: Combining local convolution with global self-attention for reading comprehension. *arXiv preprint arXiv:1804.09541*.

Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. 2017. Seqgan: Sequence generative adversarial nets with policy gradient. In *Proceddings of the AAAI*.

Rowan Zellers, Yonatan Bisk, Roy Schwartz, and Yejin Choi. 2018. Swag: A large-scale adversarial dataset for grounded commonsense inference. *arXiv preprint arXiv:1808.05326*.

Shuailiang Zhang, Hai Zhao, Yuwei Wu, Zhuosheng Zhang, Xi Zhou, and Xiang Zhou. 2019a. Dcmn+: Dual co-matching network for multi-choice reading comprehension. *arXiv preprint arXiv:1901.09381*.

Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. 2019b. Ernie: Enhanced language representation with informative entities. *arXiv preprint arXiv:1905.07129*.

Haichao Zhu, Furu Wei, Bing Qin, and Ting Liu. 2018a. Hierarchical attention flow for multiple-choice reading comprehension. In *Proceddings of the AAAI*.

Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu. 2018b. Texygen: A benchmarking platform for text generation models. In *The 41st International ACM SIGIR Conference*, pages 1097–1100. ACM.

A Appendix

A.1 Training Details

In CosmosQA experiments, we use pretrained weight of RoBERTa_large. We run experiments on a 24G Titan RTX for 5 epochs, set the max sequence length to 256. For hyper-parameters, we set the routing iterations of capsule network as 3, batch size is chosen from {8, 16, 24, 32}, learning rate is chosen from {2e-5, 1e-5, 5e-6} and warmup proportion is chosen from {0, 0.1, 0.2, 0.5}. For CEGI(F+L), the best performance is achieved at batch size=24, lr=1e-5, warmup_proportion=0.1 with 16-bit float precision. GPT2 with 12-layer and BERT_base model are used in evidence generation. In textual evidence generation, we set λ in Eq. 3 to 0.5, max sequence length to 40, batch size to 32 and the learning rate to 6.25e-0.5. In factual evidence generation, we set max sequence length to 15, batch size to 64, the learning rate to 1e-5. For both generators, we train 100000 iterations with early stop.

A.2 Baseline Methods

Cosmos Baselines

1. **Co-Matching** (Wang et al., 2018b) captures the interactions between paragraph with question and option set through attention. **Commonsense-RC** (Wang et al., 2018a) performs three-way unidirectional attention to model interactions between paragraph, question, and option set. **DMCN** (Zhang et al., 2019a) applies dual attention between paragraph and question or option set using BERT encoding output. **Multiway** (Huang et al., 2019) uses BERT to learn the semantic representation and uses multiway bidirectional interaction between each pair of input paragraph, question and option set.
2. **GPT2-FT** (Radford et al., 2018), **BERT-FT** (Devlin et al., 2018) and **RoBERTa-FT** (Liu et al., 2019) are the pretrained transformer language models with additional fine-tuning steps on CosmosQA.
3. **Commonsense-KB** (Li et al., 2019) uses logic relations from a commonsense knowledge base (e.g., ConceptNet[9]) with rule-based method to generate multiple-choice questions as additional training data to fine-tune the pretrained BERT model. **K-Adapter** (Wang et al., 2020) infuses commonsense knowledge into a large pre-trained network.

CommonsenseQA Baselines

KagNet (Lin et al., 2019) uses ConceptNet as extra knowledge and proposes a knowledge-aware graph network and finally scores answers with graph representations. **Cos-E** (Rajani et al., 2019) constructs human-annotated evidence for each question and generates evidence for test data. **DREAM** (Lv et al., 2019) adopts XLNet-large as the baseline and extracts evidence from Wikipedia. **RoBERTa + KE**, **RoBERTa + IR** and **RoBERTa + CSPT** (Lv et al., 2019) adopt RoBERTa as the baseline and utilize the evidence from Wikipedia, search engine and OMCS, respectively.

A.3 Case Study

To verify the generated evidence performance, we perform case studies on textual generator and factual generator. In addition, we also show a case that the proposed capsule network can help to select the answer by comparing with the other options.

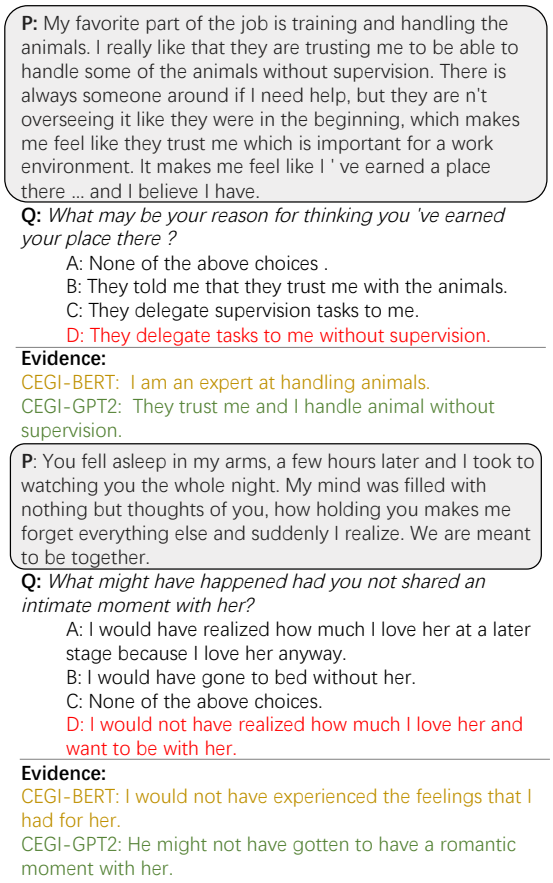

Figure 3: Examples of textual evidence generator.

Case Study on Textual Generator We show examples of automatically generated evidences by

[9] http://conceptnet.io/

CEGI-GPT2 and CEGI-BERT in Figure 3. We observe that using the multi-tasking loss, CEGI-BERT and CEGI-GPT2 generate more accurate evidence. Moreover, using those generated evidences is helpful for predicting the correct answer. In the first example, the evidence generated by CEGI-GPT2 "They trust me and I handle animal without supervision." can help select the Answer D "They delegate tasks to me without supervision." In the second example, the evidence generated by CEGI-BERT "I would not have experienced the feelings that I had for her." is close to the Answer D.

P: My nephew hates bees, and he moved over to my sisters ' house. He was trying to tell me the world did n't need bees. I told him that most plant and animal life would die within a decade of bees disappearing from the planet.

Q: *What 's a possible reason why the nephew hates bees ?*

A: Because he moved to the writer 's sister 's house.
B: Because most plant and animal life would die within decade of bees disappearing from the planet .
C: Because he got bite before.
D: None of the above choices.

Evidence:
<bee, Desires, flower> <bee, Capableof, sting> <bee, Capableof, buzz> <bee, AtLocation, any garden> <planet, HasProperty, beautiful> <planet, IsA, orbitting sun> <planet, AtLocation, solar system>, <planet, ReceivesAction, fill with sand>

P: Also, if he were to clean the interior first, he wud hafta remove the body kits and put them aside as the interior is the hardest part of any vehicle wash, having to remove oil, dirts, grease and what - nots. Soon, he stepped up to do the spray job first. Everything seemed well as he was left to spray the finished smooth surface. He began spraying right under the sun.

Q: *Why does he feel that he must perform the spray job while the sun beats down?*

A: None of the above choices.
B: The sun assists in warming the paint so it is easy to apply.
C: The sun keeps the paint from spilling off the car.
D: The sun dries the paint which is sprayed on quickly.

Evidence:
<sun, Capableof, dry something that be wet>, <spray, HasProperty, wet>, <spray Atlocation, waterfall>, <interior, HasProperty, inside>, <vehicle, UsedFor, transportation>, <vehicle, Capableof, travel>, <vehicle, UsedFor, mobility>

Figure 4: Examples of factual evidence generator.

Case Study on Factual Generator Figure 4 shows the examples of evidences generated by the factual generator. In the first example, from evidence, we know "bee is capable of sting", so option C "Because he got bite before" will be the correct answer. Some options like B "Because most plant and animal life would die within decade of bees disappearing from the planet" appear in the context "I told him that most plant and animal life would be die within a decade of bees disappearing from the planet", and thus without the evidence it could puzzle the model to select B. In the second example, we have the evidence "sun has capable of drying something that be wet" and "spray has property wet", so it is easy to reach the correct answer D "The sun dries the paint which is sprayed on quickly".

Case Study on Capsule Network We investigate the case with and without capsule network in the model. As shown in Figure 5, it is hard to answer the question simply by reading through the paragraph. However, after comparing with the other options, option A will be the best answer. In this case, the generated evidence is not useful to predict the correct answer A. But the capsule network considering all other candidate options when answering the question can help predict "She wanted her to look at a pretty rock" as answer.

P: Last night just at twilight to be exact my daughter , her little head all sweaty from running around like a little maniac with the other children in the neighborhood came bounding into the house skidding to a halt in the kitchen , \" Mommy ! Look ! Is n't it beautiful ! \"

Q: *What did her daighter want her to look at ?*

A: She wanted her to look at a pretty rock.
B: The daughter wanted her to look at the sun going down.
C: None of the above choices.
D: She wanted her to see the other pretty children.

Figure 5: Example of capsule network predict correctly while without capsule network predict wrongly.