Common Sense-Based Reasoning Using External Knowledge for Question Answering

YUNYEONG YANG AND SANGWOO KANG

1NAVER Corporation, Seongnam 13561, South Korea
2School of Computing, Gachon University, Seongnam 13120, South Korea

Corresponding author: Sangwoo Kang (swkang@gachon.ac.kr)

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No.2020-0-01907, Development of Smart Signage Technology for Automatic Classification of Untact Examination and Patient Status Based on AI).

ABSTRACT

Much research has been conducted recently on machines with human-level reasoning ability. Unlike domain knowledge, common sense is an extensive knowledge of the real world, and is included in human reasoning ability in many cases. However, it is difficult for machines to learn common sense because of the lack of well-written data. ComonsenseQA is a dataset in the form of multiple-choice questions and answers created for training and evaluating common sense in machines. RoBERTa, a pre-trained language model that performs well in natural language processing tasks, also performs well in ComonsenseQA. However, its performance is still inadequate compared to human performance. In this paper, we propose a model for predicting the correct answer by searching for information called evidence necessary to answer questions from external knowledge and by using the evidence as a context. The proposed model is comprised of three stages: An external knowledge finder to explore the required information from external knowledge; a triple-to-sentence converter, which converts triple-shaped information such as (table, AtLocation, rug) into sentence form based on the configured evidence; and a reasoning module that predicts the correct answer. The performance of our proposed model showed a performance of 80.84% on the CommonsenseQA validation data and 76.14% on the test data. The latter is an improvement of 4.04% over the baseline model RoBERTa.

INDEX TERMS
Computational and artificial intelligence, natural language processing, neural networks, machine reading comprehension.

I. INTRODUCTION

Unlike domain knowledge, common sense is a comprehensive knowledge for decision support or knowledge of the real world involving semantic understanding, such as dictionary definitions or other information. Humans identify and utilize not only the literal definitions in their knowledge but also the various relationships between different knowledge, such as the spatio-temporal and causality relations. Therefore, machines require more sophisticated reasoning techniques to utilize common sense in addition to domain knowledge to answer questions similarly to humans. A question answering system that can effectively utilize common sense will be able to answer a variety of questions as well as state simple facts like a human being.

ComonsenseQA [1] is a task that involves answering questions that require common sense not clearly expressed in the question itself. To answer the question in ComonsenseQA, the machine must reason through the implicit common sense in the question and then predict the correct answer. However, it is difficult for machines to determine the various relationships between knowledge similarly to humans. In fact, the pre-trained language model [2], [3], which shows high performance in natural language processing, also requires additional research to solve ComonsenseQA. Pre-trained language models reason using only the questions and given choices to solve ComonsenseQA. This method expects the language model to exercise common sense and predict the correct answer.

However, it still lags greatly behind human performance, and it is not known what kind of common sense the language model uses to deduce the answer.

To solve this problem, a model is proposed that creates a common sense-based context to answer the question, and predicts the correct answers based on the context. We use common sense-based contexts to fill in for the still deficient
common sense reasoning ability of the pre-trained language model. Our model can use the common sense-based context to deduce the correct answers, even though machines cannot deduce common sense on their own.

CommonsenseQA has a multiple-choice question format in which one out of five options is selected for answering the question. Figure 1 shows a question, the source concept, the choices, and the answer. The source concept is a word within the question and is the key topic of the question. One of the five options given must be chosen as the answer to the question. To answer the question, the correct answer needs to be reasoned through using common sense that does not appear within the question itself. For example, the common sense required for the question in Figure 1 is that hotels exist in cities.

There are three stages to determining the answer in our model. The first stage consists of an external knowledge finder that looks for the common sense information needed to answer the question in triple form. The second stage is a triple-to-sentence converter that converts the triple-form knowledge into sentence form. The final stage is a reasoning module that predicts the answer based on the found information. In the external knowledge finder, information is collected from ConceptNet [4], which is a representative common sense knowledge database. Information related to the question and choices is collected. This information provides evidence that each choice is the correct answer to the question. The triple-to-sentence converter converts the exploited knowledge into the form of sentences, which we call the converted text evidence. The reasoning module calculates the scores of the choices using the evidence collected for each option. We apply an attention mechanism to the evidence so that the information associated with the question can be focused on and used to predict the correct answer. The reasoning module calculates the scores for all the choices and predicts the candidate with the highest score as the correct answer.

The proposed model consisting of the external knowledge finder, triple-to-sentence converter, and reasoning module showed a 2.33% performance improvement on the CommonsenseQA validation set compared to the RoBERTa-large model and a 4.04% improvement on the test set.

II. RELATED WORK
A. COMMON SENSE
Many researchers are interested in endowing machines with human learning abilities. Machines require considerable knowledge to achieve human-level reasoning ability. In particular, common sense knowledge is difficult to conceptualize, so it is difficult for machines to learn. Common sense knowledge includes not only the knowledge of time, changes, and causality, but also various types of knowledge, including knowledge of physics, medicine, law, and social relationships. Therefore, common sense knowledge is necessary in various fields such as natural language comprehension, image understanding, and cognitive robotics. However, it is still challenging for machines to learn common sense knowledge. Because of the wide range of common sense knowledge, this problem is considered one of the most difficult problems in the field of AI research. Common sense reasoning aims to provide machines with human abilities to reason about situations in our daily life. Recently, many datasets have been created for training or evaluating machine common sense [5]–[7]. CommonsenseQA [1] is a multiple-choice QA dataset that requires different types of common sense knowledge to obtain the correct answer. We used CommonsenseQA to assess the performance of our proposed model. These datasets comprise different tasks, but the tasks have the common property that they require common sense knowledge to solve.

B. QUESTION ANSWERING

Question answering (QA) is a task in which a given question is answered. QA can generally be divided into three categories based on the answer forms, namely span-based, multiple choice, and generative. There has been considerable research on span-based QA in the QA field. Given the context and the question, this task requires the machine to extract a span of text from the context as the correct answer. The context contains not only the correct answer but also the sentences on which the correct answer is based. There is a number of datasets for span-based QA, such as SQuAD [8], TriviaQA [9], and NewsQA [10], as well as many models such as BiDAF [11], DocQA [12], and BERT [2] that perform well. In multiple-choice QA, questions and multiple options are given and the correct answer is chosen from the given options. Unlike span-based QA, context may or may not be given according to the dataset. Typical datasets include the MCTest [13], RACE [14], CommonsenseQA [1], and OpenbookQA [15]. Finally, generative QA is a task in which the correct answers are freely generated through reasoning and summary using the given questions and contexts. This task is more suitable for actual applications because there is no limit to the answers. However, it is the most complicated among all the QA tasks. The typical datasets for generative QA are MS MARCO [16], SearchQA [17], and NarrativeQA [18]. In this paper, we propose a model for solving commonsenseQA, a form of multiple-choice QA.

C. BERT

BERT [2] is a language representation model based on a multi-layer bidirectional transformer encoder. The use of BERT involves two stages: pre-training and fine-tuning.
Pre-training is first used to build a general-purpose language understanding model that uses unsupervised learning on a large text corpus. Through this unsupervised learning, BERT constructs a language model called the pre-trained BERT. Pre-trained BERT is also available as contextualized word embedding, in which different vectors are created depending on the context, even if the word remains unchanged. Fine-tuning is performed as a supervised learning task for which downstream NLP tasks such as MRC, natural language inference, and named entity recognition can be applied. For sentence classification tasks such as natural language inference and semantic analysis, the classification (CLS) token, a special token in BERT, is used for fine-tuning. The CLS vector is used to calculate the probability that a label is classified correctly. The parameters of BERT and the classification layer are fine-tuned to maximize the log probability of the correct label so that each task can be performed correctly. BERT has achieved new state-of-the-art results on 11 NLP tasks. Recently proposed variants of BERT, such as RoBERTa [3], and ALBERT [20], have demonstrated good performance on NLP tasks. In this paper, we propose a method to solve CommonsenseQA based on RoBERTa.

III. PROPOSED MODEL

We describe the overall structure of our model for solving the question answering task, which requires common sense knowledge. As shown in Figure 2, our model consists of an external knowledge finder, which brings out the common sense information needed to answer the question from external knowledge; a triple-to-sentence converter that transforms triple-form information into sentence-form to construct contexts called evidences; and the reasoning module, which predicts the correct answer based on the evidence in the constructed context.

A. EXTERNAL KNOWLEDGE FINDER

We propose an external knowledge finder to construct common sense-based contexts. To construct a context, we search for common sense information that is related to each choice and question.

Pre-training is first used to build a general-purpose language understanding model that uses unsupervised learning on a large text corpus. Through this unsupervised learning, BERT constructs a language model called the pre-trained BERT. Pre-trained BERT is also available as contextualized word embedding, in which different vectors are created depending on the context, even if the word remains unchanged. Fine-tuning is performed as a supervised learning task for which downstream NLP tasks such as MRC, natural language inference, and named entity recognition can be applied. For sentence classification tasks such as natural language inference and semantic analysis, the classification (CLS) token, a special token in BERT, is used for fine-tuning. The CLS vector is used to calculate the probability that a label is classified correctly. The parameters of BERT and the classification layer are fine-tuned to maximize the log probability of the correct label so that each task can be performed correctly. BERT has achieved new state-of-the-art results on 11 NLP tasks. Recently proposed variants of BERT, such as RoBERTa [3], and ALBERT [20], have demonstrated good performance on NLP tasks. In this paper, we propose a method to solve CommonsenseQA based on RoBERTa.

Related information refers to the K-hop paths in ConceptNet [4] that connect the source concept to the option concept. We remove unnecessary information from the paths explored to answer the question. Then, we measure the scores of the remaining paths and use the top 10 paths for the next step.

Path Exploration: To answer questions that require common sense, we need information that does not appear in the question. ConceptNet is a knowledge graph that expresses common sense in the form of various relationships between a large number of concepts, so it is not easy to find only the information needed to answer the question. To solve this problem, we propose an external knowledge finder.

The knowledge graph expressing common sense is denoted as $G$, the source concept in the question $SC$, and the i-th choice $Ci$. The source concept is a pre-determined word that is the key topic of the question in the CommonsenseQA task and is provided with each question in the dataset. We find the paths connecting the source concept $(SC)$ and the i-th choice $(Ci)$ in the knowledge graph $G$.

The discovered paths are denoted as $P_i = \{p_{i1}, p_{i2}, ..., p_{iy}\}$. The paths included inside $P_i$ are restricted to a maximum of K hops, i.e., triples connected by K edges. Figure 3 shows the hop paths with less than 3 hops connecting “Table” and “Rug”. In this case, the path “Table → Rug” is a one-hop path because it consists of one edge. “Table ← Crumbs → Rug” is a two-hop path because it consists of triples with two edges, and “Table → Floor → Crumbs → Rug” is a three-hop path. In the proposed model, we use paths with two or fewer hops.

Consider the example of the choice “a sense of calm” and the question “What type of feeling is performing for the first time likely to produce?” in which the source concept is “performing”. However, “a sense of calm” does not exist in ConceptNet as a concept, so it cannot be searched for directly. We therefore search for “sense” and for “calm”, which is the stem of a “sense of calm”. Then, we search for paths connecting “sense” and “performing”, and for paths connecting “calm” and “performing”. We denote the path sets explored for all the choices $C$ as $P = \{P_1, P_2, ..., P_3\}$.

However, the paths linking the source concept to the option $Ci$ contain information that is redundant in answering the questions. For example, the “table ← dog → rug” path in Figure 3 does not support obtaining the answer to the question...
"If I keep getting crumbs under my table, what should I put under it?" These unnecessary paths should be removed from the search results. We assess a path that does not include the words in the question as being unrelated to the question. In Figure 3, "table ← dog → rug" is removed, as the word in the path other than the source concept and the choice, i.e., "dog", does not exist in the question.

We then calculate the scores for all the paths \( \{p_{i1}, p_{i2}, \ldots, p_{ij}\} \) belonging to the set of paths \( P_i \). The measured score indicates how much information required to answer the question is present in the path. The scores are calculated using the following formulae:

\[
\text{Score} (p_{ij}) = \frac{T\left((\text{Question} \cup \text{Choice}) \cap \text{Path}_{ij}\right)}{T(\text{Path}_{ij})} \quad (1)
\]

\[
T(x) = n(\text{stem}(x)) \quad (2)
\]

The scores are the percentage of words in the path that also appear in the question and the choice. The denominator of the score represents the number of stems in the concepts present in the \( j \)-th path. The numerator of the score represents the number of stems in the concepts present in the question and the choice. The denominator of the path other than the source concept and the choice, i.e., "dog", does not exist in the question.

We then calculate the scores for all the paths \( \{p_{i1}, p_{i2}, \ldots, p_{ij}\} \) belonging to the set of paths \( P_i \). The measured score indicates how much information required to answer the question is present in the path. The scores are calculated using the following formulae:

\[
\text{Score} (p_{ij}) = \frac{T\left((\text{Question} \cup \text{Choice}) \cap \text{Path}_{ij}\right)}{T(\text{Path}_{ij})} \quad (1)
\]

\[
T(x) = n(\text{stem}(x)) \quad (2)
\]

The scores are the percentage of words in the path that also appear in the question and the choice. The denominator of the score represents the number of stems in the concepts present in the \( j \)-th path. The numerator of the score represents the number of stems in the concepts present in the question and the choice. The denominator of the path other than the source concept and the choice, i.e., "dog", does not exist in the question.

### B. TRIPLE-TO-SENTENCE CONVERTER

The top ten explored paths \( \{p_{i1}, p_{i2}, \ldots, p_{ij}\} \) are in the form of several triples. They must be converted to sentence form to be used as input in the next step, i.e., the reasoning module. At this point, we refer to the context consisting of the top ten paths converted into sentences as "evidence". Each explored path \( p_{ij} \) consists of several triples. We convert each triple into one-sentence form and attach the [EOT] token, which denotes the end of the triple. Each evidence therefore consists of several sentences separated by [EOT] tokens. (Table, AtLocation, Rug) can be replaced by a sentence such as "Table at location rug". However, this sentence is not grammatically correct. Grammatically incorrect sentences are difficult for machines to understand [19]. Therefore, we use templates to convert triples into sentences. There are several candidates in the template for the relationship "AtLocation". Table 1 shows examples of candidate sentences formed using the templates. Among the candidate sentences in Table 1 are inappropriate sentences such as "Somewhere table can be is rug."

We use the RoBERTa language model to choose the most natural sentence among the various candidate sentences. We evaluate the appropriateness of candidate sentences converted using templates via the language model because the probability that a natural or grammatically correct sentence is appropriate is higher than the probability that an unnatural or ungrammatical sentence is appropriate.

Each of the several triples existing on the path \( p_{ij} \) is converted to \( T_{ij} = \{T_{ij}^1, T_{ij}^2, T_{ij}^3, \ldots, T_{ij}^m\} \) using a template. For

### C. REASONING MODULE

Our proposed reasoning module takes the questions, choices, and the previously constructed evidence as input. Based on the given input, the reasoning module calculates how appropriate each choice is as the answer to the question. We designed the model to base its reasoning on the evidence obtained from the external knowledge finder to calculate the plausibility score for which the answer to the question is a choice. We calculate the score for all the choices and predict the choice with the highest score as the correct answer. The reasoning module consists of an encoding layer, a GRU layer, an attention layer, and an output layer. The encoding layer is based on RoBERTa and is used for contextualized word embedding for the input comprising the question, the choices, and the evidence. The GRU layer is used to obtain more contextual information about the inputs. The attention layer concentrates on sentences in the evidence that contain the information needed to answer the question. Finally, the output layer uses the evidence to calculate the score for each choice.

#### 1) ENCODING LAYER

We use RoBERTa [3], which is a pre-trained language model, to obtain the contextualized word embeddings of questions, choices, and evidence. The input has the form of "<s> question + choice </s> Evidence </s>". The input sentences are divided into tokens by the tokenizer, and the token embedding that reflects the context of each token uses the output \( t_i \) of the last hidden layer in RoBERTa.

| TABLE 1. Examples of candidate sentences for the triple. |
|-------------------|------------------------------------------|
| Triple             | Candidate sentences \( T_i^* \)             |
| Table, AtLocation, Rug | "You are likely to find table in rug" |
|                    | "Something you find on rug is table"      |
|                    | "Something you find under rug is table"   |
|                    | "You are likely to find table at rug"     |
|                    | "Something you find in rug is table"      |
|                    | "Some where table can be is rug"          |
|                    | "Something you find under rug is table"   |

\( T_{ij} \), the sentence \( T_{ij}^* \) with the highest probability of being a correct sentence calculated by the language model is selected. The probability of a sentence is calculated by multiplying the probabilities of every token that forms the sentence \( \{w_1, w_2, w_3, \ldots, w_N\} \):

\[
P(t) = \prod_{k=0}^{N} P(w_k|w_0, \ldots w_{k-1}, w_{k+1}, \ldots, w_N),
\]

\[
T_{ij}^* = \arg \max_{t \in T_i} \{P(t)\}. \quad (4)
\]

All the triples in the top ten paths \( p_{ij}^* \) are converted to sentences as described above. Each sentence is separated by EOT and used as the evidence \( E_i \) in the reasoning module.
2) GRU LAYER
We employ the RNN architecture to obtain more contextual information about the input. Among the various types of RNNs, we chose GRU to recognize contextual information. GRU takes the embedding \( t_i \) of each token, which is the output of the encoding layer, as input. GRU then outputs a presentation \( h_i \) that includes the relational information obtained for each input token which is later used in the attention layer.

3) ATTENTION LAYER
The attention layer is used to focus on the critical part of the evidence during the reasoning. We identify the multi-sentence evidence that is the most relevant to the question and give weight to the critical part. Attention is used as shown in Figure 4 to determine how the question and choices relate to each sentence in the evidence. The equations used for the attention in this study are:

\[
QA = \frac{1}{q} \sum_{i=0}^{q} h_i
\]  
\[
k_n = \frac{1}{n(S_n)} \sum_{i \in S_n} h_i
\]  
\[
e' = [QA^T k_1, QA^T k_2, \ldots, QA^T k_n]
\]  
\[
\alpha' = \text{softmax} \left( e' \right)
\]  
\[
KB = \sum_{j=1}^{N} \alpha'_j kb_j
\]  

4) OUTPUT LAYER
Finally, the output layer calculates a score for how appropriate each choice is as the answer to the question. The score is calculated as \( \text{softmax}(W^T [QA; KB]) \). We calculate the scores for all the choices. The predicted answer is the choice that has the highest score.

IV. EXPERIMENTS
We performed experiments to verify the effectiveness of the proposed model. We present a description of CommonsenseQA, the dataset used, the implementation details of the experiments, the experimental results, and the ablation analysis.

A. DATASET SPECIFICATION
We evaluated the proposed model using the CommonsenseQA dataset [1]. CommonsenseQA is a question answering dataset for training machines to learn common sense. A question answering dataset that provides the context does not require any common sense to solve because the given context provides a basis for the correct answer. However, because CommonsenseQA does not provide the context, the necessary information must be obtained from common sense knowledge. CommonsenseQA has a multiple-choice structure in which the answer to each question is selected from the given 5 choices. CommonsenseQA is a dataset generated by crowd workers based on ConceptNet. The author of CommonsenseQA selected source concepts from ConceptNet and, for each source concept, three target concepts with the same relationship to the source concept. These concepts are provided to the crowd workers. For each of the target concepts, the crowd workers provided a corresponding question for which the source concept is the key topic of the question and the target concept is the correct answer while the other two target concepts constitute two of the other four choices. The fourth choice for each question is another concept that has the same relationship to the source concept as the target concepts. The last choice is a word related to the question but not the answer determined by the crowd workers.

CommonsenseQA is a more difficult dataset because the choices have the same relationship to the key topic in the question. The CommonsenseQA dataset consists of 12,102 question-option pairs. The entire dataset was divided into training (9,741), validation (1,221), and test (1,140) datasets. The accuracy was used as the evaluation metric. The human performance on CommonsenseQA is 88.9%.

B. TRAINING DETAILS
We used the RoBERTa-large model as the encoding layer of the reasoning module. The RoBERTa-base model was used to calculate the probabilities of the sentences generated in the triple-to-sentence converter. The hyperparameters selected after fine-tuning the reasoning module using the CommonsenseQA dataset are shown in Table 2. We confirmed that the best performance for the validation set occurs at 12000 steps (4 epochs), and that the best performance is 76.41% on the test set.

C. EXPERIMENT RESULTS
We show the performance results achieved by the proposed model in the experiment. The performance of the proposed model on the validation set is 80.83%, which is an improvement of about 2.33% compared to the RoBERTa-large model used as the baseline in this study. The proposed model achieved an improvement of 4.04%. We performed a comparative experiment with a few other models.
TABLE 2. Hyperparameters used in the fine-tuning.

| Hyperparameter   | Value   |
|------------------|---------|
| Batch size       | 3       |
| Max sequence length | 256   |
| Epoch            | 4       |
| Learning rate    | 3e-6    |
| Optimizer        | Adam    |
| Dropout          | 0.1     |
| Weight decay     | 0.01    |
| GRU size         | 768     |

TABLE 3. Model performance on CommonsenseQA.

| Group  | Model             | Validation Acc | Test Acc |
|--------|-------------------|---------------|---------|
|        | Human             | 88.9          |         |
| Proposed model | 80.83 | 76.14         |         |
| Group 1 | RoBERTa          | 78.5          | 72.1    |
|         | XLNET            | -             | 62.9    |
|         | BERTLarge        | -             | 56.7    |
|         | BERTBase         | -             | 53.0    |
| Group 2 | HyKAS 2.0        | 80.1          | 73.2    |
|         | RoBERTa + IR     | 78.9          | 72.1    |
|         | RoBERT + KE      | 78.5          | 68.4    |
|         | KagNet (RoBERTa) | 77.56         | -       |
| Group 3 | XLNET + graph reasoning | 79.3 | 75.3 |
|         | HyKAS 2.0        | 80.1          | 73.2    |
|         | KagNet (RoBERTa) | 77.56         | -       |
|         | KagNet (BERT)    | 64.46         | 58.9    |

1) MODEL PERFORMANCE FOR COMMONSENSEQA
Table 3 shows the performance of the models on the CommonsenseQA validation and test sets. The proposed model demonstrated a performance of 80.83% for the validation set and a performance of 76.14% for the test set. We divided the models compared into three groups.

Group 1 comprises the RoBERTa [3], XLNET [12], and BERT [2] models that fine-tune the pre-trained language model with CommonsenseQA. The input to these models takes the form of "<s> question </s> choice </s>".

The models were fine-tuned to predict the score of the choice using the encoding vector of the first input, <s>. The pre-trained language models show good performance after fine-tuning using only the questions and choices without external knowledge. However, they still do not perform as well as humans. Our proposed model solves these problems with a 2.33% improvement on the validation set and a 4.04% improvement on the test set compared to the baseline model RoBERTa-large.

Group 2 comprises models that use additional external knowledge while using RoBERTa as the encoding layer, similar to the proposed model. KagNet [22], and HyKAS 2.0 [23] use ConceptNet as the external knowledge source, similar to the proposed model. RoBERTa + KE uses wiki docs as the external knowledge source to search for sentences while RoBERTa + IR uses the Open Mind Common Sense (OMCS) [24] and searches for sentences using a search engine. Our model showed the best performance among the models using various external knowledge sources.

Group 3 includes models that use ConceptNet for external knowledge. KagNet, XLNET + Graph Reasoning [25], and HyKAS 2.0 use ConceptNet, similar to the proposed model, but with different model structures to predict the correct answer. The proposed model showed the highest performance among the models using ConceptNet. The comparison with the major models shows that the proposed model is effective for commonsenseQA.

2) EFFECTS OF NUMBER OF HOPS
We conducted an experiment in which the number of hops was varied. The results are shown in Table 4. It is difficult to find the right answer for CommonsenseQA with a one-level relation, i.e., a one-hop path. This is because four of the options have the same relationship between the source concept of the question and the answer. In addition, as shown in Figure 3, the answer to the question requires multiple hop relations. We therefore performed an experiment to find the optimal number of hops (K). The results of the experiment demonstrate that using 2 hops provides the best performance. We found that the performance with 3 and 4 hops is lower than that with 1 hop. This result shows that the inclusion of unnecessary information interferes with the model’s prediction of the correct answer.

3) EFFECT OF RNN TYPES
Table 5 displays the results of the experiment in which the type of RNN layer used on the contexts in the reasoning module were varied. A RNN is a model that processes sequential data in which the previous results affect the states that follow. This property makes the use of a RNN appropriate for identifying the context of the input. The types of RNN used in the experiment are the vanilla RNN, LSTM, and GRU. The LSTM is an improved model for solving the vanishing gradient problem present in the vanilla RNN. The GRU is a simplified LSTM model but has similar performance to LSTM.

TABLE 4. Performance with different RNN types.

| Hops (K) | Validation Acc |
|---------|---------------|
| 1       | 80.01         |
| 2       | 80.83         |
| 3       | 79.52         |
| 4       | 78.78         |
A unidirectional RNN has limitations in expressing the relationships with the previous words. To solve this problem, a bidirectional RNN with forward and backward directions was proposed. We used the type of RNN that showed the highest performance in the comparative experiments between the different types of RNN layers in the model. The results proved that the unidirectional RNN resulted in the best performance.

### D. ABLATION STUDY

We conducted an ablation study to check the effectiveness of each component in the proposed model. Table 6 shows the results of the ablation study. The data used for the ablation experiment is the CommonsenseQA validation set.

First, we used all the paths searched in the external knowledge finder without removing the unnecessary information to answer the questions. The results show that there was a drop of about 1.5% in the performance and that the proposed pruning method improves the performance. We also confirmed that the split search with stems for the choice or source concepts that do not exist as concepts in ConceptNet improves the performance.

We then removed the attention and GRU in the reasoning module. The experiments confirm that attention contributes significantly to the performance of our model, and that identifying the contextual information of each token using GRU also improves the performance.

### V. CONCLUSION

In this paper, we proposed a model to predict the correct answers using a common sense-based context to solve questions that require common sense. An external knowledge finder was designed to locate the common sense knowledge necessary to answer the questions. The triple-to-sentence converter converts paths consisting of triples found using the external knowledge finder to evidence consisting of sentences. The evidence is then used as common sense-based context for calculating the score of each prospective correct answer in the reasoning module. RoBERTa, a pre-trained language model, is used in the reasoning module to obtain an embedding that reflects the context of the input. We use a GRU layer to obtain contextual information, and an attention layer to focus on the information related to the question in the evidence. Finally, the output layer calculates how appropriate each choice is as the answer to the question.

The performance of the model proposed in this paper was confirmed using the CommonsenseQA dataset. The proposed model achieved 80.83% accuracy on the validation data and 76.16% accuracy on the test data. Our performance was better by 2.53% on the validation data and 4.04% on the test data compared to the baseline RoBERTa-large model. Moreover, we analyzed the impact of each component of the proposed model on the performance improvement. For future research, we plan to use ALBERT, a pre-trained language model that performs better than RoBERTa as an encoding layer.

### REFERENCES

[1] A. Talmor, J. Herzig, N. Lourie, and J. Berant, “CommonSenseQA: A question answering challenge targeting commonsense knowledge,” in *Proc. NAACL*, Minneapolis, MN, USA, 2019, pp. 4149–4158.

[2] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,” in *Proc. NAACL*, Minneapolis, MN, USA, 2019, pp. 4171–4186.

[3] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, “RoBERTa: A robustly optimized BERT pretraining approach,” 2019, arXiv:1907.11692. [Online]. Available: http://arxiv.org/abs/1907.11692

[4] R. Speer, J. Chin, and C. Havasi, “Conceptnet 5.5: An open multilingual graph of general knowledge,” in *Proc. 31st AAAI Conf. Artif. Intell.*, San Francisco, CA, USA, 2017, pp. 4444–4451.

[5] R. Zellers, Y. Bisk, A. Farhadi, and Y. Choi, “From recognition to cognition: Visual commonsense reasoning,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Long Beach, CA, USA, Jun. 2019, pp. 6713–6724.

[6] R. Zellers, Y. Bisk, R. Schwartz, and Y. Choi, “SWAG: A large-scale adversarial dataset for grounded commonsense inference,” in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2018, pp. 93–104.

[7] H. Levesque, E. Davis, and L. Morgenstern, “The winograd schema challenge,” in *Proc. Int. Conf. Knowl. Repres. Reasoning (KR)*, Tempe, AZ, USA, 2012, pp. 552–561.

[8] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, “SQuAD: 100,000+ questions for machine comprehension of text,” in *Proc. Conf. Empirical Methods Natural Lang. Process.*, Austin, TX, USA, 2016, pp. 2383–2392.

[9] M. Joshi, E. Choi, D. Weld, and L. Zettlemoyer, “TriviaQA: A large-scale distantly supervised challenge dataset for reading comprehension,” in *Proc. 55th Annu. Meeting Assoc. Comput. Linguistics*, Vancouver, BC, Canada, vol. 1, 2017, pp. 1601–1611.

[10] A. Trischler, T. Wang, X. Yuan, J. Harris, A. Sordoni, P. Bachman, and K. Suleman, “NewsQA: A machine comprehension dataset,” in *Proc. 2nd Workshop Represent. Learn. NLP*, Vancouver, BC, Canada, 2017, pp. 191–200.

[11] M. Seo, A. Kembhavi, A. Farhadi, and H. Hajishirzi, “Bidirectional attention flow for machine comprehension,” in *Proc. ICLR*, Toulon, France, 2017, pp. 1–14.

[12] R. Yang, J. Zhang, X. Gao, F. Ji, and H. Chen, “Simple and effective text matching with richer alignment features,” in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics*, Florence, Italy, 2019, pp. 4699–4709.

[13] M. Richardson, C. J. C. Burges, and E. Renshaw, “MCTest: A challenge dataset for the open-domain machine comprehension of text,” in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, Seattle, WA, USA, 2013, pp. 193–203.

[14] G. Lai, Q. Xie, H. Liu, Y. Yang, and E. Hovy, “RACE: Large-scale reading comprehension dataset from examinations,” in *Proc. Conf. Empirical Methods Natural Lang. Process.*, Copenhagen, Denmark, 2017, pp. 785–794.
YUNYEONG YANG received the bachelor’s degree in computer engineering from Kwangwoon University and the master’s degree in computer science and engineering, by writing a thesis on Question Answering, from Sogang University, in 2020. She is currently a Researcher with the Department of Natural Language Processing, NAVER Corporation. Her research interests include machine reading comprehension, question answering, sequence labeling, and information extraction.

SANGWOO KANG received the Ph.D. degree in computer science from Sogang University. He was a Research Fellow Professor with Sogang University. Since September 2016, he has been an Assistant Professor with the School of Computing, Gachon University, where he is currently leading the Natural Language Processing Laboratory. He is specialized in natural language processing and interested in spoken dialogue interface, information retrieval, text mining, opinion mining, big data, and UI/UX. His research interest includes applying deep learning techniques to his research.

* * *