Article

Considering Commonsense in Solving QA: Reading Comprehension with Semantic Search and Continual Learning

Seungwon Jeong 1,†, Dongsuk Oh 2,†, Kinam Park 1 and Heuiseok Lim 2,*

1 Human-Inspired AI & Computing Research Center, Korea University, Seoul 02841, Korea; redlion0929@korea.ac.kr (S.J.); spknn@korea.ac.kr (K.P.)
2 Department of Computer Science and Engineering, Korea University, Seoul 02841, Korea;
inow3555@korea.ac.kr
* Correspondence: limhseok@korea.ac.kr; Tel.: +82-2-3290-2396
† These authors contributed equally to this work.

Abstract: Unlike previous dialogue-based question-answering (QA) datasets, DREAM, multiple-choice Dialogue-based REAding comprehension exaMination dataset, requires a deep understanding of dialogue. Many problems require multi-sentence reasoning, whereas some require commonsense reasoning. However, most pre-trained language models (PTLMs) do not consider commonsense. In addition, because the maximum number of tokens that a language model (LM) can deal with is limited, the entire dialogue history cannot be included. The resulting information loss has an adverse effect on performance. To address these problems, we propose a Dialogue-based QA model with Common-sense Reasoning (DQACR), a language model that exploits Semantic Search and continual learning. We used Semantic Search to complement information loss from truncated dialogue. In addition, we used Semantic Search and continual learning to improve the PTLM's commonsense reasoning. Our model achieves an improvement of approximately 1.5% over the baseline method and can thus facilitate QA-related tasks. It contributes toward not only dialogue-based QA tasks but also another form of QA datasets for future tasks.

Keywords: dialogue-based multiple-choice QA; commonsense reasoning; semantic search; pre-trained language models; deep learning

1. Introduction

Machine reading comprehension (MRC) is a technology by which a machine can find answers to questions about a given document. RACE [1] and SQuAD [2] are passage-based reading comprehension datasets. Using such datasets, a machine can learn to find answers from a given passage. In contrast to these datasets, DREAM [3] is a dialogue-based question-answering (QA) dataset that focuses on in-depth multi-turn multi-party dialogue understanding. It consists of 6444 dialogues and 10,197 questions. Each data item consists of one dialogue, one question, three candidates, and one answer. DREAM presents more challenges than previous dialogue-based multiple-choice QA datasets. Specifically, 84% of the answers are non-extractive, 85% of the questions require multi-sentence reasoning, and 34% of the questions require commonsense reasoning. A sample from DREAM is presented in Table 1. Thus, a high level of commonsense reasoning and a deep understanding of dialogue are required to improve performance. Therefore, we propose a Dialogue-based QA model with Common-sense Reasoning (DQACR).

Fine-tuning of pre-trained language models (PTLMs) using dialogue-based QA datasets has been shown to be effective [4,5]. However, this method has several drawbacks. If the input sequence is longer than the number of tokens that the model can take, the rear part of the dialogue history becomes truncated. A PTLM contains a fixed number of tokens that can be received. For example, BERT [6] can receive up to 512 tokens. Such truncation can result in information loss, with subsequent degradation of performance owing to the...
inadequate input of important information. Furthermore, because PTLMs use only the given dialogue history to find the answers to questions, it is difficult to implement them for solving problems that require commonsense reasoning.

Table 1. A sample question in DREAM dataset. To answer this question, commonsense is required to explain the necessity of thorough cleaning after a party.

| Dialogue | W: Forgive my mess. We had a party last night. A lot of people came over and they all brought food and drinks. M: Yeah, I can tell. Well, I think it’s pretty obvious what you’ll be doing today. |
|----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Question | What will the woman probably do today?                                                                                                                                                             |
| Candidate Answer | (a) Get more food and drinks. (b) Make a thorough cleaning. (√) (c) Ask her friends to come over.                                                                                                       |

Using Semantic Search (SS), DQACR identifies sentences relevant to the questions within the dialogue history. We can reduce the information loss caused by truncation if the selected sentences are used as the input instead of the entire dialogue history. However, problems remain with the commonsense reasoning of the model.

Commonsense is an inherent trait of humans. However, machines cannot acquire commonsense without learning-related knowledge [7]. Continual learning and Semantic Search can improve the commonsense reasoning of a machine. Continual learning is a method in which a PTLM is fine-tuned with a task required to perform the current task in advance. By pre-training a model with CommonsenseQA [8] (CSQA), a typical commonsense inference task, we can improve the commonsense reasoning of the model. ConceptNet [9] can also be used to improve commonsense reasoning. From Refs. [10,11], we know that such an approach can help improve commonsense reasoning performance. Considering ConceptNet knowledge about the problem, the model can read a dialogue with commonsense. This process can be useful in terms of gaining a deep understanding of dialogue and improving commonsense reasoning.

The main contributions of this study are as follows:

- Commonsense reasoning of a PTLM can be improved by learning commonsense through continual learning.
- Semantic Search is used to reduce the information loss in the dialogue history and improve commonsense reasoning using ConceptNet.
- DQACR achieves better performance than the baseline method.

2. Related Work

This section discusses Pre-Trained Language Model, Semantic Search, and continual learning. In addition, it deals with ConceptNet and CommonsenseQA.

2.1. Pre-Trained Language Model (PTLM)

Natural language processing (NLP) research with deep learning is widely studied in various fields [12,13]. Recently, the active undertaking of research using a Pre-Trained Language Model (PTLM) has been carried out. PTLM represents a language model (LM) that has been trained with a large dataset to learn an appropriate way to represent language. In pre-training, the LM performs unsupervised learning using a large corpus without labeling. One of the most commonly used PTLMs, BERT [6] was pre-trained using the Masked Language Model (MLM) and Next Sentence Prediction (NSP) tasks. After pre-training, the LM is fine-tuned with a specific dataset to perform the corresponding task. This method has the advantage of providing good performance even with relatively small amounts of data. In the case of BERT, it achieved state-of-the-art on the GLUE benchmark [14], SQuAD 1.1 [2], SQuAD 2.0 [15], and SWAG [16].
2.2. Semantic Search

Semantic Search is a method for finding sentences that are semantically similar to target sentences on the basis of similarity. For Semantic Search, sentences with similar meanings are mapped close to the latent space. The similarity between sentences can be computed through the cosine similarity, dot product, etc., between vectors. In contrast to the TF-IDF scheme [17,18], this method can consider the latent meaning of a sentence. Sentence transformers (https://www.sbert.net/docs/pretrained_models.html, accessed on 1 March 2022) represent a Semantic Search framework based on a PTLM and provide pre-trained models. Among them, the checkpoint ‘all-mpnet-base-v2’ [19] shows the highest average performance. Therefore, we implemented this model in our Semantic Search.

2.3. ConceptNet

Humans use commonsense to understand sentences semantically. ConceptNet [9] is a knowledge graph format that contains information about common words and phrases as well as commonsense relationships. Each node represents a word or phrase in a sentence, and an edge represents a commonsense relationship between two nodes. Knowledge in ConceptNet is collected from various sources such as Open Mind Common Sense (OMCS) [20], Open Multilingual WordNet [21], and DBPedia [22]. Each data item is represented by <entity1, relation, entity2>. For example, in the case of entity1:young, entity2:senior, relation:Antonym, young is the antonym of senior. Since a model that learns ConceptNet considers commonsense, it can understand the semantic meaning of sentences more deeply.

2.4. CommonsenseQA

CommonsenseQA [8] is multiple-choice QA dataset for learning and improving the commonsense reasoning of a model. Data are generated using concepts extracted from ConceptNet. Crowd workers use the concepts to construct multiple-choice QA. Each data item consists of one question, five candidate answers, and one answer. For each problem, the target concept must be distinguished, and the candidate answers are confusing concepts. A dataset generated in this manner helps ensure a clear understanding of commonsense. PTLMs such as ELMO [23] and BERT [6] show low performance in commonsense reasoning, whereas XLNet [24], RoBERTA [25], and ALBERT [26] show high performance. In particular, ALBERT-based commonsense reasoning models show the highest performance (https://www.tau-nlp.org/csqa-leaderboard2, accessed on 1 March 2022).

2.5. Continual Learning

To learn multiplication, we must first learn addition. Learning addition before multiplication leads to a deeper understanding and better performance. A language model (LM) is based on a similar approach. If Task A influences the model to learn Task B, it is better to learn Task A first and then learn Task B. This is called continual learning. Continual learning is currently used in many areas of NLP. ERNIE 2.0 [27] uses continual learning in the process of pre-training. In addition, continual learning can be used to carry out tasks in dialogue systems [28] and named entity recognition (NER) [29] with good performance.

3. Method

The proposed DQACR model includes the CSQA and DREAM modules. The left and right parts of Figure 1 show the CSQA and DREAM modules, respectively. First, the PTLM is fine-tuned with CommonsenseQA [8] in the CSQA module. This improves the commonsense reasoning ability of the PTLM. Next, we create the model input from DREAM [3] using ConceptNet [9] SS, and dialogue SS in the DREAM module. Finally, we adopt continual learning, in which the model that learned CommonsenseQA is fine-tuned with the modified input from the previous implementation.
3.1. CommonsenseQA Fine-Tuning

Humans understand the semantic meaning of context on the basis of the commonsense that they have acquired over their lives. However, without such external knowledge, a machine acquires only a shallow understanding of the context. If a machine can acquire commonsense-related knowledge, it will be able to gain a deep understanding of sentences in the appropriate context. When a PTLM learns a QA dataset related to commonsense, the parameters of the model are adjusted to improve commonsense reasoning. Thus, the model that learns CommonsenseQA, a typical commonsense-related QA dataset, has the advantage of commonsense reasoning over the PTLM. In fine-tuning, the input is in the form ‘[CLS] question [SEP] candidate answer [SEP]’. Cross-entropy loss of the following form is used:

\[
\text{loss}_{csqa} = -\log\left(\frac{\exp(L[y])}{\sum_{i=1}^{5} \exp(L[i])}\right)
\]  

where \( L \) denotes the hidden representation from the last layer of the model and \( y \) denotes the label. Because the number of candidate answers of CommonsenseQA is five, the range of adding in the denominator is expressed as \( i = 1 \) to 5. Based on Equation (1), PTLM proceeds with learning in the direction of improving commonsense reasoning.

3.2. ConceptNet Semantic Search (SS)

In general, a PTLM uses only the dialogue history given to solve the problem; hence, it is difficult to use the PTLM to solve a problem that requires commonsense. For the PTLM
to consider commonsense, external commonsense must be used as well [7]. Given this requirement, the model can refer to the relevant commonsense when reading the dialogue history. We applied Semantic Search to ConceptNet, one of the most used commonsense datasets, to extract the commonsense knowledge most relevant to each candidate answer. Semantic Search is used on the basis of the cosine similarity between each candidate answer and each ConceptNet data item. The cosine similarity is mathematically expressed as follows:

$$sim(c^k, o^l) = \frac{c^k \cdot o^l}{\|c^k\|\|o^l\|}$$  \hspace{1cm} (2)

where \(c^k\) denotes the embedding of \(k\)th sentence from ConceptNet and \(o^l\) denotes the embedding of \(l\)th candidate answer. The higher the similarity between the ConceptNet data item and the candidate answer, the greater is the value of Equation (2). After the most similar data item is found, it is concatenated at the beginning of the input. Therefore, the input is in the form ‘[CLS] \(c^{optimal}\) [SEP] dialogue history [SEP] question \(o^l\) [SEP]’.

3.3. Dialogue Semantic Search (SS)

Because the input sequence length that PTLM can process at one time is limited, dialogue history with a length that is greater than this limit cannot be used. Therefore, we present a method for effectively using dialogue history within a limited sequence length. We eliminate utterances that have low relevance to the question from the dialogue until the input sequence length reaches the maximum capacity of the model. This process ensures that the most relevant information can be used to solve the problem. Semantic Search is used on the basis of the cosine similarity between each dialogue utterance and the question. The cosine similarity is mathematically expressed as follows:

$$sim(u^k, q) = \frac{u^k \cdot q}{\|u^k\|\|q\|}$$  \hspace{1cm} (3)

where \(u^k\) denotes the embedding of \(k\)th utterance in the dialogue history and \(q\) denotes the embedding of a question. The higher the similarity between the question and the utterance, the greater the value of Equation (3). Based on the value of Equation (3), the least similar utterances are removed one by one. While making the input, the order of each utterance is maintained to preserve the contextual flow. This method can minimize information loss and enable the model to use as many relevant utterances as possible.

3.4. DREAM Fine-Tuning

For the PTLM to perform a specific task, fine-tuning with the specific dataset is necessary. Since we want to make the PTLM perform the DREAM task, we fine-tune the PTLM with the DREAM dataset. When fine-tuning the PTLM with DREAM, the parameters of the model are adjusted in the direction specified by the dialogue-based multiple-choice QA task. Specifically, it learns how to solve problems based on the dialogue history. In fine-tuning, we use the following cross-entropy Loss:

$$loss_{dream} = -\log\left(\frac{\exp(L[y])}{\sum_{i=1}^{3} \exp(L[i])}\right)$$  \hspace{1cm} (4)

where \(L\) denotes the hidden representation from the last layer of the model and \(y\) denotes the label. Because the number of candidate answers of DREAM is three, the range of adding in the denominator is expressed as \(i = 1\) to \(3\). Based on Equation (4), the PTLM proceeds with learning in the direction of improving the DREAM task.
4. Experiments

4.1. Experimental Setup

4.1.1. Data

DREAM [3] is a dialogue-based multiple-choice reading comprehension dataset. Each data item consists of one dialogue history, one question, three candidate answers, and one answer. Further, 34% of the questions require commonsense reasoning. Information on the configuration of the training, development, and test datasets is presented in Table 2.

Table 2. Training, development, and test set division of DREAM.

|           | Train | Dev  | Test | All  |
|-----------|-------|------|------|------|
| # of dialogues | 3869  | 1288 | 1287 | 6444 |
| # of questions | 6116  | 2040 | 2041 | 10,197 |

To apply Semantic Search, the triple <entity1, relation, entity2> in ConceptNet [9] is transformed into a sentence “entity1 relation entity”. Because the dialogue history is selected as a Semantic Search for questions, ConceptNet knowledge is selected as a Semantic Search considering the candidate answers. The ConceptNet used is version 5.7 (https://github.com/commonsense/conceptnet5/wiki, accessed on 1 March 2022).

4.1.2. Parameters

When learning the DREAM task, we use ALBERT$_{\text{xlarge}}$ [26] and RoBERTa$_{\text{large}}$ [25] as the PTLM. The parameters used for fine-tuning DREAM are listed in Table 3. In Semantic Search (https://www.sbert.net/docs/pretrained_models.html, accessed on 1 March 2022), a pre-trained sentence transformer, ‘all-mpnet-base-v2’ [19] (https://huggingface.co/sentence-transformers/all-mpnet-base-v2, accessed on 1 March 2022), is used owing to its best average performance. When learning CommonsenseQA [8], we use the model that is fine-tuned with CommonsenseQA (https://huggingface.co/danlou/albert-xxlarge-v2-finetuned-csqa, https://huggingface.co/danlou/roberta-large-finetuned-csqa, accessed on 1 March 2022) in advance. The following parameters are employed: learning rate, $1 \times 10^{-5}$; train batch size, 16; eval batch size, 16; seed, 42; optimizer, Adam with beta = (0.9, 0.999) and epsilon = $1 \times 10^{-8}$; lr scheduler type, linear; num epochs, 5; mixed precision training, Native AMP.

Table 3. Hyperparameters when fine-tuning the DREAM dataset.

| Parameter             | Value     |
|-----------------------|-----------|
| Train batch size      | 9         |
| Eval batch size       | 9         |
| Epoch                 | 3         |
| Weight decay          | 0.01      |
| Learning rate         | $1 \times 10^{-5}$ |

4.2. Analysis of Experimental Results

In this section, we demonstrate the effectiveness of each strategy. Table 4 summarizes the overall experimental results, i.e., the results obtained by adding each of dialogue SS, continual learning, and ConceptNet SS. We achieved a performance improvement of 1.5% over the baseline method.
Table 4. Model performance for DREAM. The base model employs ALBERT\textsubscript{xxlarge}. The modules used are denoted by √ in each line. DSS stands for dialogue Semantic Search, CSS stands for ConceptNet Semantic Search, and CSQA CL stands for CommonsenseQA Continual Learning. We achieved a performance improvement of 1.5% over the baseline method.

| Model                | DSS | CSS | CSQA CL | Accuracy (%) |
|----------------------|-----|-----|---------|--------------|
| ALBERT\textsubscript{xxlarge} | √   | √   | √       | 87.90        |
|                      | √   |     | √       | 88.68        |
| DQACR (ALBERT\textsubscript{xxlarge}) | √   | √   | √       | 90.05        |

4.2.1. Dialogue Semantic Search (SS)

This section demonstrates the effectiveness of dialogue SS. We removed the sentence that was least relevant to the question in order to configure the input in the maximum capacity of the model. This can reduce the information loss caused by truncation. Table 4 shows that a performance improvement of 0.33% over the baseline method is achieved when dialogue SS is applied. In addition, when dialogue SS was removed from our model, the performance degraded by 1.37%, i.e., from 90.05% to 88.68%. This reduction is due to truncation of the rear part of the dialogue history when its length is greater than the maximum token length that the model can receive.

Table 5 shows the dialogue when dialogue SS is not used, and Table 6 shows the dialogue when dialogue SS is used. We compare these two tables to demonstrate the effectiveness of dialogue SS. Since the dialogue is longer than the model can deal with, the dialogue is truncated at the last utterance of the boy in Table 5. In this problem, our baseline ALBERT\textsubscript{xxlarge} received the dialogue in Table 5 as input, not the entire dialogue. The model deduces the correct answer by referring to the part that is not truncated. Hence, it selects the answer as (c) by referring to the previous utterance of the father, i.e., “butterflies flying around the zoo”. It cannot use the appropriate information because the information required to solve the problem has been truncated. Our baseline ALBERT\textsubscript{xxlarge} selected the wrong answer in this problem. However, Table 6 consists of information that is highly relevant to the problem. In this problem, the model using dialogue SS received the dialogue in Table 6 that is relevant to the question. Therefore, the model can find the appropriate information to solve the problem. The model used the boy’s utterance “they’re inside” as well as the next utterance “What was it made of? [Glass]” to infer the correct answer as “(a) inside a glass enclosure”. Dialogue SS can thus capture the information required to solve the problem while minimizing information loss. Therefore, we conclude that if the length of dialogue is longer than the model can deal with, dialogue SS can effectively improve the performance.

Table 5. Dialogue history without applying dialogue SS. Because the latter part is truncated, the model cannot use the related information during the inference process. This adversely affects the inference of the model.

| Dialogue                                                                 |
|--------------------------------------------------------------------------|
| Father: Mikey. Time for bed [Why?] Why? It’s getting dark out. Well, do you want to talk before you go to bed? [Yeah]  |
| Boy: Uh, what do you want to talk about?                                 |
| ...                                                                      |
| Father: Uh, okay. Then, you saw some butterflies, did not you? [Yeah]   |
| What colors were they?                                                  |
| Boy: After the bird show.                                               |
| Father: After the bird show you saw them. Furthermore, were the butterflies flying around all over the zoo? |
| Boy: Uh, um, they’re ← truncate                                         |

Table 6. Dialogue history with applying dialogue SS. The model can use the appropriate information that is relevant to the question, and the inference process is less affected.
Table 5. Cont.

| Question | Where did the boy see the butterflies? |
|----------|-----------------------------------------|

| Candidate answer |
|------------------|
| (a) inside a glass enclosure |
| (b) in a wire building near the bird show |
| (c) flying around the zoo |

Table 6. Dialogue history when applying dialogue SS. Since this is a summary of the dialogue related to the question, the model can use appropriate information in the inference process. This is effective because the model employs as much useful information as possible.

Dialogue

Father: Uhh, okay. Then, you saw some butterflies, did not you? [Yeah] What colors were they?
Boy: After the bird show.
Father: After the bird show you saw them. Furthermore, were the butterflies flying around all over the zoo?
Boy: Uh, um, they’re inside.
Father: They were inside, what, a little building? [Yeah]
What was the building made of? Was it made of wood? [No]
What was it made of? [Glass] Oh, made of glass.
Furthermore, could not the butterflies fly out of the glass? [No] No, oh, what stopped them from flying out?
Boy: Um, the air.
Father: Oh, the air. Oh, there was air coming down? [Yeah]
Oh, well that is great. Well, it’s time to go to bed now.
Sleep tight and do not let the bed bugs bite. Good night.

Question

Where did the boy see the butterflies?

Candidate answer

(a) inside a glass enclosure
(b) in a wire building near the bird show
(c) flying around the zoo

4.2.2. CSQA Continual Learning

Here, we discuss the effectiveness of continual learning in solving the problem. Many problems in DREAM require commonsense reasoning. Solving these problems contributes toward achieving a high score. To solve multiplication, the learning of addition must occur first. Similarly, learning about commonsense is necessary to solve problems requiring commonsense. Thus, we sequentially trained CommonsenseQA and DREAM in PTLM. Continual learning proved a very effective method for improving commonsense reasoning of the model. Table 4 shows that a performance improvement of 0.57% over the baseline method is achieved when continual learning is applied. These improvements show that the model trained with CommonsenseQA actually solves the problems by considering commonsense. In addition, removing CSQA continual learning from DQACR reduced the performance by 2.15%. This shows that CSQA continual learning has the greatest effect in terms of performance improvement.

4.2.3. ConceptNet Semantic Search (SS)

If we add ConceptNet knowledge related to the candidate answer to the input, the model can refer to the knowledge related to the problem. As shown in the third and fifth lines of Table 4, the performance is degraded compared to the baseline method when ConceptNet SS is applied. If the commonsense reasoning is below a certain level, this method adversely affects the performance. Because adding ConceptNet information reduces the length of the dialogue that can be used, the model cannot employ appropriate information to solve problem. However, applying ConceptNet SS to models with dialogue SS and CSQA continual learning improves the performance by 0.29%. Therefore, we conclude that applying external commonsense to models with a certain level of commonsense reasoning can help improve performance.
4.3. Experimental Results of Other LMs

Here, we demonstrate that our implementation is also effective in the case of other LMs. We choose RoBERTa\textsubscript{large}, which shows high performance on DREAM and CommonsenseQA. As can be seen in Table 7, our implementation also improved the performance of RoBERTa\textsubscript{large}. Thus, it can be concluded that our idea is efficient not only for ALBERT but also for other LMs in terms of improving their ability to solve DREAM problems.

Table 7. DREAM performance on ALBERT\textsubscript{xxlarge} and RoBERTa\textsubscript{large}.

| Model                   | Acc  |
|-------------------------|------|
| Vanilla ALBERT\textsubscript{xxlarge} | 88.50 |
| DQACR (ALBERT\textsubscript{xxlarge})  | 90.05 |
| Vanilla RoBERTa\textsubscript{large}  | 84.66 |
| DQACR (RoBERTa\textsubscript{large})   | 85.64 |

5. Conclusions and Future Works

Dialogue-based multiple-choice QA tasks using existing PTLMs bear the following disadvantages: (1) Limited length of dialogue history that can be entered as input and (2) insufficient ability to perform commonsense reasoning. Through Semantic Search, we improved truncation-related problems by employing only relevant sentences as input. Moreover, we improved commonsense reasoning using CommonsenseQA \cite{8} continual learning and ConceptNet \cite{9} Semantic Search. Thus, we achieved a performance improvement of approximately 1.5% over the baseline method. In addition, our model contributes toward not only dialogue-based QA tasks but also QA datasets for future tasks, such as RACE \cite{1} and SQuAD \cite{2}.

However, our model has the following drawbacks: (1) It is overly dependent on Semantic Search results. If a ConceptNet sentence found through Semantic Search does not help solve the problem, it can reduce the amount of dialogue history used by the model and thus degrade the overall performance. (2) Although Semantic Search reduces information loss, some loss still occurs because of the truncated dialogue history. Therefore, in the future, we will carry out a study to improve these drawbacks. In total, 66% of the problems in the DREAM dataset can be solved without commonsense. Employing ConceptNet SS for such a problem rather reduces the length of dialogue that the model can refer to for solving the problem. The sole application of ConceptNet SS to problems requiring commonsense reasoning will further improve the performance. Thus, we will investigate a method in which the problem can be solved more efficiently by combining it with a classifier that determines whether the problem requires commonsense or not. Therefore, we will study the method used for the LM to solve the problem more efficiently by combining a classifier that distinguishes whether the problem requires commonsense or not.

Author Contributions: Conceptualization, software, investigation, methodology, visualization, writing—review and editing—S.J. and D.O.; investigation, visualization, writing—original draft—K.P.; validation, supervision, resources, project administration, and funding acquisition, H.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2021R1A6A1A03045425). In addition, it was also supported by the MSIT (Ministry of Science and ICT), Korea, under the ICT Creative Consilience program (IITP-2022-2020-0-01819) supervised by the IITP (Institute for Information & communications Technology Planning & Evaluation).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.
Data Availability Statement: DREAM: https://dataset.org/dream/, accessed on 1 March 2022, CommonsenseQA: https://www.tau-nlp.org/commonsenseqa, accessed on 1 March 2022, ConceptNet: https://conceptnet.io/, accessed on 1 March 2022.

Conflicts of Interest: The authors declare no conflict of interest.

References
1. Lai, G.; Xie, Q.; Liu, H.; Yang, Y.; Hovy, E. Race: Large-scale reading comprehension dataset from examinations. arXiv 2017, arXiv:1704.04683.
2. Rajpurkar, P.; Zhang, J.; Lopyrev, K.; Liang, P. Squad: 100,000+ questions for machine comprehension of text. arXiv 2016, arXiv:1606.05250.
3. Sun, K.; Yu, D.; Chen, J.; Yu, D.; Choi, Y.; Cardie, C. Dream: A challenge data set and models for dialogue-based reading comprehension. Trans. Assoc. Comput. Linguist. 2019, 7, 217–231. [CrossRef]
4. Zhao, Y.; Zhang, Z.; Zhao, H. Reference knowledgeable network for machine reading comprehension. arXiv 2020, arXiv:2012.03709.
5. Jin, D.; Gao, S.; Kao, J.Y.; Chung, T.; Hakkani-Tur, D. Mm: Multi-stage multi-task learning for multi-choice reading comprehension. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 8010–8017.
6. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv 2018, arXiv:1810.04805.
7. Lim, J.; Oh, D.; Jang, Y.; Yang, K.; Lim, H. I know what you asked: Graph path learning using AMR for commonsense reasoning. arXiv 2020, arXiv:2011.00766.
8. Talmor, A.; Herzig, J.; Lourie, N.; Berant, J. Commonsenseqa: A question answering challenge targeting commonsense knowledge. arXiv 2018, arXiv:1811.00937.
9. Speer, R.; Chin, J.; Havasi, C. Conceptnet 5.5: An open multilingual graph of general knowledge. In Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017.
10. Xu, Y.; Zhu, C.; Xu, R.; Liu, Y.; Zeng, M.; Huang, X. Fusing context into knowledge graph for commonsense question answering. arXiv 2020, arXiv:2012.04808.
11. Yan, J.; Raman, M.; Chan, A.; Zhang, T.; Rossi, R.; Zhao, H.; Kim, S.; Lipka, N.; Ren, X. Learning Contextualized Knowledge Structures for Commonsense Reasoning. arXiv 2020, arXiv:2010.12873.
12. Daashtipour, K.; Gogate, M.; Li, J.; Jiang, F.; Kong, B.; Hussain, A. A hybrid Persian sentiment analysis framework: Integrating dependency grammar based rules and deep neural networks. Neurocomputing 2020, 380, 1–10. [CrossRef]
13. Ke, W.; Gao, J.; Shen, H.; Cheng, X. ConsistSum: Unsupervised Opinion Summarization with the Consistency of Aspect, Sentiment and Semantic. In Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining, Phoenix, AZ, USA, 21–25 February 2022; pp. 467–475.
14. Wang, A.; Singh, A.; Michael, J.; Hill, F.; Levy, O.; Bowman, S.R. GLUE: A multi-task benchmark and analysis platform for natural language understanding. arXiv 2020, arXiv:1804.07461.
15. Rajpurkar, P.; Jia, R.; Liang, P. Know what you do not know: Unanswerable questions for SQuAD. arXiv 2018, arXiv:1806.03822.
16. Zellers, R.; Bisk, Y.; Schwartz, R.; Choi, Y. Swag: A large-scale adversarial dataset for grounded commonsense inference. arXiv 2018, arXiv:1808.05326.
17. Joachims, T. A Probabilistic Analysis of the Rocchio Algorithm with TFIDF for Text Categorization; Technical Report; Department of Computer Science, Carnegie Mellon University: Pittsburgh, PA, USA, 1996.
18. Robertson, S.; Zaragoza, H. The Probabilistic Relevance Framework: BM25 and Beyond; Now Publishers Inc.: Delft, The Netherlands, 2009.
19. Song, K.; Tan, X.; Qin, T.; Lu, J.; Liu, T.Y. Mpnet: Masked and permuted pre-training for language understanding. Adv. Neural Inf. Process. Syst. 2020, 33, 16857–16867.
20. Singh, P.; Lin, T.; Mueller, E.T.; Lim, G.; Perkins, T.; Zhu, W.L. Open mind common sense: Knowledge acquisition from the general public. In OTM Confederated International Conferences “On the Move to Meaningful Internet Systems”; Springer: Berlin/Heidelberg, Germany, 2002; pp. 1223–1237.
21. Bond, F.; Foster, R. Linking and extending an open multilingual wordnet. In Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Sofia, Bulgaria, 4–9 August 2013; pp. 1352–1362.
22. Auer, S.; Bizer, C.; Kobilarov, G.; Lehmann, J.; Cyganiak, R.; Ives, Z. Dbpedia: A nucleus for a web of open data. In The Semantic Web; Springer: Berlin/Heidelberg, Germany, 2007; pp. 722–735.
23. Peters, M.E.; Neumann, M.; Iyyer, M.; Gardner, M.; Clark, C.; Lee, K.; Zettlemoyer, L. Deep Contextualized Word Representations. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), New Orleans, LA, USA, 1–6 June 2018; pp. 2227–2237.
24. Yang, Z.; Dai, Z.; Yang, Y.; Carbonell, J.; Salakhutdinov, R.R.; Le, Q.V. Xlnet: Generalized autoregressive pretraining for language understanding. In Proceedings of the Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, Vancouver, BC, Canada, 8–14 December 2019; Volume 32.
25. Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; Stoyanov, V. Roberta: A robustly optimized bert pretraining approach. arXiv 2019, arXiv:1907.11692.
26. Lan, Z.; Chen, M.; Goodman, S.; Gimpel, K.; Sharma, P.; Soricut, R. Albert: A lite bert for self-supervised learning of language representations. *arXiv* 2019, arXiv:1909.11942.

27. Sun, Y.; Wang, S.; Li, Y.; Feng, S.; Tian, H.; Wu, H.; Wang, H. Ernie 2.0: A continual pre-training framework for language understanding. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 8968–8975.

28. Madotto, A.; Lin, Z.; Zhou, Z.; Moon, S.; Crook, P.; Liu, B.; Yu, Z.; Cho, E.; Wang, Z. Continual learning in task-oriented dialogue systems. *arXiv* 2020, arXiv:2012.15504.

29. Monaikul, N.; Castellucci, G.; Filice, S.; Rokhlenko, O. Continual learning for named entity recognition. In Proceedings of the Thirty-Fifth AAAI Conference on Artificial Intelligence, Vancouver, BC, Canada, 2–9 February 2021; pp. 13570–13577.