Improving Question Answering by Commonsense-Based Pre-Training

Wanjun Zhong[1*], Duyu Tang[2], Nan Duan[2], Ming Zhou[2], Jiahai Wang[1], and Jian Yin[1]

[1] The School of Data and Computer Science, Sun Yat-sen University. Guangdong Key Laboratory of Big Data Analysis and Processing, Guangzhou, P.R.China
[2] Microsoft Research Asia, Beijing, China
{zhongwj25@mail2,wangjiah@mail,issjyin@mail}.sysu.edu.cn
{dutang,nanduan,mingzhou}@microsoft.com

Abstract. Although neural network approaches achieve remarkable success on a variety of NLP tasks, many of them struggle to answer questions that require commonsense knowledge. We believe the main reason is the lack of commonsense connections between concepts. To remedy this, we provide a simple and effective method that leverages external commonsense knowledge base such as ConceptNet. We pre-train direct and indirect relational functions between concepts, and show that these pre-trained functions could be easily added to existing neural network models. Results show that incorporating commonsense-based function improves the state-of-the-art on three question answering tasks that require commonsense reasoning. Further analysis shows that our system discovers and leverages useful evidence from an external commonsense knowledge base, which is missing in existing neural network models and help derive the correct answer.

1 Introduction

Commonsense reasoning is a major challenge for question answering [9, 4, 16, 2]. Take Figure 1 as an example. Answering both questions requires a natural language understanding system that has the ability of reasoning based on commonsense knowledge about the world. Although neural network approaches have

Id	Question	Candidate Answers
1	Which element makes up most of the air we breathe?	(A) carbon (B) nitrogen (C) oxygen (D) argon
2	Which property of a mineral can be determined just by looking at it?	(A) luster (B) mass (C) weight (D) hardness

Fig. 1. Examples from ARC [4] that require commonsense knowledge and reasoning.

* Work is done during internship at Microsoft Research Asia.

achieved promising performance when supplied with a large number of supervised training instances, even surpassing human-level exact match accuracy on the Stanford Question Answering Dataset (SQuAD) benchmark [18], it has been shown that existing systems lack true language understanding and reasoning capabilities [7], which are crucial to commonsense reasoning. Moreover, although it is easy for humans to answer the questions mentioned above based on their knowledge about the world, it is a great challenge for machines when there is limited training data.

In this paper, we leverage external commonsense knowledge, such as ConceptNet [20], to improve the commonsense reasoning capability of a question answering (QA) system. We believe that a desirable way is to pre-train a generic model from external commonsense knowledge about the world, with the following advantages. First, such a model has a broader coverage of the concepts/entities and can access rich contexts from the relational knowledge graph. Second, the ability of commonsense reasoning is not limited to the number of training instances and the coverage of reasoning types in the end tasks. Third, it is convenient to build a hybrid system that preserves the semantic matching ability of the existing QA system, which might be a neural network-based model, and further integrates a generic model to improve model's capability of commonsense reasoning.

We believe that the main reason why the majority of existing methods lack the commonsense reasoning ability is the absence of connections between concepts[3]. These connections could be divided into direct and indirect ones. Below is an example sampled from ConceptNet. In this case, {"*driving*", "*a license*"}

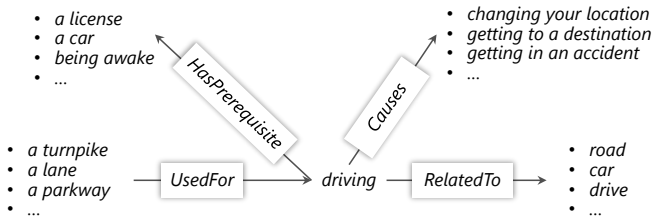

Fig. 2. A sampled subgraph from ConceptNet with "*driving*" as the central word.

forms a direct connection whose relation is "*HasPrerequisite*". {"*driving*", "*road*"} also forms a direct connection. Moreover, there are indirect connections here such as {"*a car*", "*getting to a destination*"}, which are connected by a pivot concept "*driving*". Based on this, people can learn two functions to measure direct and indirect connections between every pair of concepts. These functions could be easily combined with existing QA system to make decisions.

We take three question answering tasks [4, 16, 12] that require commonsense reasoning as the testbeds. These tasks take a question and optionally a con-

[3] In this work, concepts are words and phrases that can be extracted from natural language text [20].

text[4] as input, and select an answer from a set of candidate answers. We believe that understanding and answering the question requires knowledge of both words and the world [6]. Thus, we implement document-based neural network based baselines and use the same way to improve the baseline systems with our commonsense-based pre-trained models. Results show that incorporating pre-trained models brings improvements on these three tasks and improve model's ability to discover useful evidence from an external commonsense knowledge base.

The first contribution of our work is that we present a simple yet effective way to pre-train commonsense-based functions to capture the semantic relationships between concepts. The pre-training model can be easily incorporated into other tasks requiring commonsense reasoning. Secondly, we demonstrate that incorporating the pre-trained model improves strong baselines on three multi-choice question answering datasets.

2 Tasks and Datasets

Given a question of length M and optionally a supporting passage of length N, both tasks are to predict the correct answer from a set of candidate answers. The difference between these tasks is the definition of the supporting passage which will be described later in this section. Systems are expected to select the correct answer from multiple candidate answers by reasoning out the question and the supporting passage. Following previous studies, we regard the problem as a ranking task. At the test time, the model should return the answer with the highest score as the prediction.

The **first** task comes from SemEval 2018 Task 11[5] [16], which aims to evaluate a system's ability to perform commonsense reasoning in question answering. The dataset describes events about daily activities. For each question, the supporting passage is a specific document given as a part of the input, and the number of candidate answers is two.

The **second** task we focus on is ARC, short for AI2 Reasoning Challenge, proposed by [4][6]. The ARC Dataset consists of a collection of scientific questions and a large scientific text corpus containing a large number of science facts. Each question has multiple candidate answers (mostly 4-way multiple candidate answers). The dataset is separated into an easy set and a challenging set. The Challenging Set contains only difficult, grade-school questions including questions answered incorrectly by both a retrieval-based algorithm and a word co-occurrence algorithm, and have acquired strong reasoning ability of commonsense knowledge or other reasoning procedure [2]. Figure 1 shows two examples which need to be solved by common sense. We target at the challenge set here.

The **third** dataset we use in the experiment is OpenBook QA[7], which calls for exploring the knowledge from an open book fact and commonsense knowledge

[4] The definitions of contexts in these tasks are slightly different and we will describe the details in the next section.

[5] https://competitions.codalab.org/competitions/17184

[6] http://data.allenai.org/arc/arc-corpus/

[7] http://data.allenai.org/OpenBookQA

from other sources. [12]. The dataset consists of 5,957 multiple-choice questions (4,957/500/500 for training/validation/test) and a set of 1,326 facts about elementary level science.

3 Commonsense Knowledge

This section describes the commonsense knowledge base we investigate in our experiment. We use ConceptNet[8] [20], one of the most widely used commonsense knowledge bases. Our approach is generic and could also be applied to other commonsense knowledge bases such as WebChild [21], which we leave as future work. ConceptNet is a semantic network that represents the large sets of words and phrases and the commonsense relationships between them. It contains 657,637 instances and 39 types of relationships. Each instance in ConceptNet can be generally described as a triple $r_i = (subject, relation, object)$. For example, the "*IsA*" relation (e.g. "*car*", "*IsA*", "*vehicle*") means that "*XX is a kind of YY*"; the "*Causes*" relation (e.g. "*car*", "*Causes*", "*pollution*") means that "*the effect of XX is YY*"; the "*CapableOf*" relation (e.g. "*car*", "*CapableOf*", "*go fast*") means that "*XX can YY*", etc. More relations and explanations could be found at [20].

4 Approach Overview

In this section, we give an overview of our framework to show the basic idea of solving the commonsense reasoning problem. Details of each component will be described in the following sections.

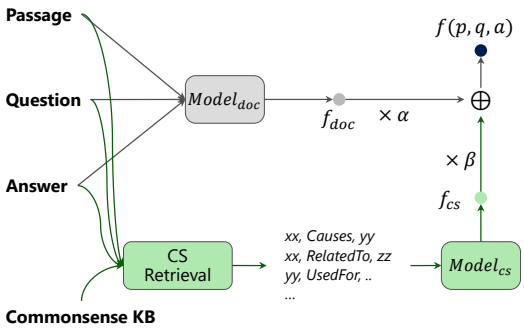

Fig. 3. An overview of our system for commonsense based question answering.

At the top of our framework, we suggest that we should select the candidate answer with the highest probability (highest score) as our final prediction. So we

[8] http://conceptnet.io/

can tackle this problem by designing a scoring function that captures the evidence mentioned in the passage and retrieved from the commonsense knowledge base.

An overview of the QA system is given in Figure 3. We define the scoring function $f(a_i)$ to calculate the score of a candidate answer a_i, which can be calculated by the sum of document based scoring function $f_{doc}(a_i)$ and commonsense based scoring function $f_{cs}(a_i)$.

$$f(a_i) = \alpha f_{doc}(a_i) + \beta f_{cs}(a_i) \tag{1}$$

The calculation of the final score would consider the given passage, the given question, and a set of commonsense knowledge related to this instance.

In the next section we will detail the design and mathematical formulas of our commonsense knowledge based scoring function. Due to the page limit, we put the description on the document-based model in the appendix.

5 Commonsense-based Model

In this section, we first describe how to pre-train commonsense-based functions to capture the semantic relationships between two concepts. Graph neural network [19] is used to integrate context from the graph structure in an external commonsense knowledge base. Afterward, we present how to use the pre-trained functions to calculate the relevance score between two pieces of text, such as a question sentence and a candidate answer sentence.

We model both direct and indirect relations between two concepts from commonsense KB, both of which are helpful when the connection between two sources (e.g., a question and a candidate answer) is missing based on the word utterances merely. Take direction relation involved in Figure 4 as an example.

Question	Candidate Answers
Why does a plastic rod have a negative charge after being rubbed with a piece of fur	(A) The fur gives up protons to the rod (B) The rod gives up electrons to the air (C) The fur gains protons from the rod (D) The rod gains electrons from the fur

Fig. 4. An example from ARC dataset. The analysis of this example could be improved if it is given the fact {"*electrons*", "*HasA*", "*negative charge*"} in ConceptNet.

If a model is given the evidence from ConceptNet such that the concept "*electrons*" and the concept "*negative charge*" has direct relation, it would be more confident to distinguish between (B,D) and (A,C), thus has a larger probability of obtaining the correct answer (D). Therefore, it is desirable to model the relevance between the two concepts. Moreover, ConceptNet could not cover all the concepts which potentially have direction relations. We need to model the direct relation for every two concepts.

Similarly, indirect relation also provides strong evidence for prediction making. As shown in the example of Fig 2, the concept "*a car*" has an indirect

relation to the concept "*getting to a destination*", both of which have a direct connection to the pivot concept "*driving*". With access to this information, a model would give a higher score to the answer containing "*car*" when questioned "*how did someone get to the destination*". Therefore, we model the commonsense-based relation between two concepts c_1 and c_2 as follows, where $\odot$ means element-wise multiplication, $Enc(c)$ stands for an encoder that represents a concept c with a continuous vector.

$$f_{cs}(c_1, c_2) = Enc(c_1) \odot Enc(c_2) \tag{2}$$

Specifically, we represent a concept with two types of information, namely the words it contains and the neighbors connected to it in the structural knowledge graph. From the first aspect, since each concept might consist of a sequence of words, we encode it by a bidirectional LSTM over Glove word vectors [17], where the concatenation of hidden states at both ends is used as the representation. We denote it as $h^w(c) = BiLSTM(Emb(c))$. From the second aspect, we represent each concept based on the representations of its neighbors and the relations that connect them. We get inspirations from graph neural network [19]. We regard a relation that connects two concepts as the compositional modifier to modify the meaning of the neighboring concept. Matrix-vector multiplication is used as the composition function [15]. We denote the neighbor-based representation of a concept c as $h^n(c)$, which is calculated as follows, where $r(c, c')$ is the specific relation between two concepts, $NBR(c)$ stands for the set of neighbors of the concept c, W and b are model parameters.

$$h^n(c) = \sum_{c' \in NBR(c)} (W^{r(c,c')} h^w(c') + b^{r(c,c')}) \tag{3}$$

The final representation of a concept c is the concatenation of both representations, namely $Enc(c) = [h^w(c); h^n(c)]$.

We use a standard ranking-based loss function to train the parameters, which is given in Equation 4.

$$l(c_1, c_2, c') = max(0, f_{cs}(c_1, c') - f_{cs}(c_1, c_2) + mgn) \tag{4}$$

In this equation, c_1 and c_2 form a positive instance, which means that they have a relationship with each other, while c_1 and c' form a negative instance. mgn is the margin with value of 0.1 in the experiment. We can easily learn two functions to model direct and indirect relations between two concepts by having different definitions of what a positive instance is, and accordingly using different strategies to sample the training instances. For the direct relation, we set those directly adjacent entities pairs in the knowledge graph as positive examples and randomly select entity pairs that have no direct relationship as negative examples. For the indirect relation, we select entity pairs that have a common neighbor as a positive instance and randomly select an equal number of entities pairs that have no one-hop or two-hop connected relations as negative instances.

We denote the direct relation based function as $f_{cs}^{dir}(c_1, c_2)$, and the indirect relation based function as $f_{cs}^{ind}(c_1, c_2)$. The final commonsense-based score in Equation 1 is calculated by using one of these two functions, or using both of them through a weighted sum. We will show the results under different settings in the experiment section.

We detailed the commonsense-based functions to measure the direct and indirect connection of each pair of concepts. Here, we present how to calculate the commonsense based score of a question sentence and a candidate answer sentence. In our experiment, we retrieve commonsense facts from ConceptNet [20]. As described above, each fact from ConceptNet can be represented as a triple, namely $c = (subject, relation, object)$. For each sentence (or paragraph), we retrieve a set of facts from ConceptNet. Specifically, we first extract a set of the n-grams from each sentence. We experiment with $\{1, 2, 3\}$-gram in our searching process, and then, we save the commonsense facts from ConceptNet which contain one of the extracted n-grams. We denote the facts for a sentence s as E_s.

Suppose we have obtained commonsense facts for a question sentence and a candidate answer, respectively, let us denote the outputs as E_1 and E_2. We can calculate the final score by the following formula. The intuition is to select the most relevant concept of each concept in E_1, and then aggregate all these scores by average.

$$f_{cs}(a_i) = \frac{1}{|E_1|} \sum_{x \in E_1} \max_{y \in E_2} (f_{cs}(x, y)) \tag{5}$$

In the experiments, we also apply the previous scoring function for a pair of paragraph and candidate answer, where E_1 and E_2 come from the supporting paragraph and the answer sentence, respectively. Furthermore, we also calculate an additional $f_{cs}(a_i)$ score for the answer-paragraph pair in the same way. For a paragraph-question pair, to guarantee the relevance of the candidate answer sentence, we filter out concepts from E_1 or E_2, if they are not contained in the extracted concepts from the candidate answer.

Our method differs from TransE [3] in three aspects. Firstly, the goals are different. The goal of TransE is to embed entities and predicates/relations into low-dimensional vector space. Secondly, the outputs are different. TransE outputs embeddings of entities and predicates, while out model outputs the parameterized scoring function. Thirdly, the evidence used for representing entities are different. Compared to TransE, our model further incorporates the neighbors of concepts via graph neural network.

6 Experiment

We conduct experiments on three question answering datasets, namely SemEval 2018 Task 11 [16], ARC Challenge Dataset [4] and OpenBook QA Dataset [12] to evaluate the effectiveness of our system. To improve the generality of our model, we trained the document based model and commonsense based model separately, which can make the commonsense based model easier to be incorporated into other tasks. We report model comparisons and model analysis in this section.

6.1 Model Comparisons and Analysis

On ARC, SemEval and OpenBook QA datasets, we follow existing studies and use accuracy as the evaluation metric. Table 1 and Table 2 show the results on these three datasets, respectively. On the ARC and OpenBook QA dataset, we compare our model with a list of existing systems. On the SemEval dataset, we only report the results of TriAN, which is the top-performing system in the SemEval evaluation[9]. f_{cs}^{dir} is our commonsense-based model for direct relations, and f_{cs}^{ind} represents the commonsense-based model for indirect relations. From the results, we can observe that commonsense-based scores improve the accuracy of the document-based model TriAN, and combining both scores could achieve further improvements on both datasets. The results show that our commonsense-based models are complementary to standard document-based models. We also apply BERT [5] to improve our baseline and show our method enhance the performance on the stronger baseline. The details of applying BERT will be explained in the appendix.

Table 1. Performances of different approaches on the the ARC Challenge dataset (left), and OpenBook QA dataset (right). F indicates the golden fact for the question.

Model	Accuracy
IR	20.26%
TupleInference	23.83%
DecompAttn	24.34%
Guess-all	25.02%
DGEM-OpenIE	26.41%
BiDAF	26.54%
Table ILP	26.97%
DGEM	27.11%
KG^2	31.70%
BiLSTM Max-out	33.87%
ET-RR	36.36%
TriAN	31.25%
TriAN + f_{cs}^{dir}	32.28%
TriAN + f_{cs}^{ind}	32.96%
TriAN + f_{cs}^{dir} + f_{cs}^{ind}	33.39%
TriAN(Concat Bert)	35.18%
TriAN(Concat Bert)+f_{cs}^{dir} + f_{cs}^{ind}	36.55%

Model	Accuracy
NO TRAINING, F+KB	
IR	24.8%
TupleInference	26.6%
DGEM	24.6%
PMI	21.2%
TRAINED MODELS, NO F or KB	
Embedd+Sim	41.8%
ESIM	48.9%
PAD	49.6%
Odd-one-out Solver	50.2%
Question Match	50.2%
ORACLE MODELS, F AND/OR KB	
f	55.8%
f + WordNet	56.3 %
f + ConceptNet	53.7 %
TriAN	56.6%
TriAN + f_{cs}^{dir} + f_{cs}^{ind}	58.0%
TriAN + BERT	70.6%
TriAN + BERT+ f_{cs}^{dir} + f_{cs}^{ind}	72.8%

Figure 5 shows an example from SemEval that benefits from both direct and indirect relations from commonsense knowledge. Despite both the question and candidate (A) mention about "*drive/driving*", the document-based model fails to make the correct prediction. We can see that the retrieved facts from ConceptNet help from different perspectives. The fact {"*driving*","*HasPrerequisite*","*license*"} directly connects the question to the candidate (A), and both {"*license*", "*Synonym*", "*permit*"} and {"*driver*","*RelatedTo*","*care*"} directly connects candidate (A) to the passage. Besides, we calculate for the question-passage pair,

[9] During the SemEval evaluation, systems including TriAN report results based on model pretraining on RACE dataset [8] and system ensemble. In this work, we report numbers on SemEval without pre-trained on RACE or ensemble.

Table 2. Performances of different approaches on the SemEval Challenge dataset.

Model	Accuracy
TriAN	80.33%
TriAN + f_{cs}^{dir}	81.58%
TriAN + f_{cs}^{ind}	81.44%
TriAN + f_{cs}^{dir} + f_{cs}^{ind}	81.80%
TriAN + BERT	86.27%
TriAN + BERT+ f_{cs}^{dir} + f_{cs}^{ind}	87.49%

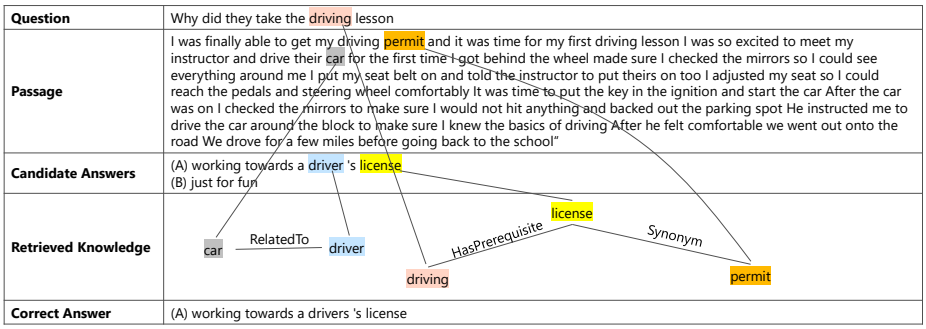

Fig. 5. An example from SemEval 2018 that requires sophistic reasoning based on commonsense knowledge.

where the indirect relation between {"*driving*","*permit*"} could be used as side information for prediction.

We further make comparisons by implementing different strategies to use commonsense knowledge from ConceptNet. We implement three baselines, including **TransE** [3], Pointwise Mutual Information (**PMI**) and Key-Value Memory Network (**KV-MemNet**) [14]. Detailed descriptions about these baselines can be found at the appendix. From Table 3 we can see that learning direct

Table 3. Performances of approaches with different strategies to use commonsense knowledge on ARC, SemEval 2018 Task 11 and OpenBook QA datasets.

Model	ARC	SemEval	OBQA
TriAN	31.25%	80.33%	56.6%
TriAN + PMI	31.72%	80.50%	53.1%
TriAN + TransE	30.59%	80.37%	55.2%
TriAN + KV-MemNet	30.49%	80.59%	54.6%
TriAN + f_{cs}^{dir} + f_{cs}^{ind}	33.39%	81.80%	58.0%

and indirection connections based on contexts from word-level constituents and neighbor from knowledge graph performs better than TransE which is originally designed for KB completion. PMI performs well, however, its performance is limited by the information it can take into account, i.e. the word count infor-

mation. The comparison between KV-MemNet and our approach further reveals the effectiveness of pretraining.

6.2 Error Analysis and Discussion

We analyze the wrongly predicted instances from both datasets and summarize the majority of errors of the following groups.

The first type of error, which is also the dominant one, is caused by failing to highlight the most useful concept in all the retrieved ones. The usefulness of a concept should also be measured by its relevance to the question, its relevance to the document, and whether introducing it could help distinguish between candidate answers. For example, the question is "*Where was the table set*" is asked based on a document talking about dinner, according to which two candidate answers are "*On the coffee table*" and "*At their house*". Although the retrieved concepts for the first candidate answer also being relevant, they are not relevant to the question type "*where*". We believe that the problem would be alleviated by incorporating a context-aware module to model the importance of a retrieved concept in a particular instance and combining it with the pre-trained model to make the final prediction.

The second type of error is caused by the ambiguity of the entity/concept to be linked to the external knowledge base. For example, suppose the document talks about computer science and machine learning, the concept "*Micheal Jordan*" in question should be linked to the machine learning expert rather than the basketball player. However, to achieve this requires an entity/concept disambiguation model, the input of which also considers the question and the passage.

Moreover, the current system fails to handle difficult questions which need logical reasoning, such as "*How long do the eggs cook for*" and "*How many people went to the movie together*". We believe that deep question understanding, such as parsing a question based on a predefined grammar and operators in a semantic parsing manner [10], is required to handle these questions, which is a promising direction, and we leave it to future work.

7 Related Work

Current top-performing methods in MRC datasets are dominated by neural models. Our commonsense-based model, which is pre-trained on commonsense KB, is complementary to this line of work and has proven effective in two question answering tasks through model combination. Our work relates to recent neural network approaches that incorporate side information from external and structured knowledge bases [1]. Existing studies roughly fall into two groups, where the first group aims to enhance each basic computational unit (e.g., a word or a noun phrase) and the second group aims to support external signals at the top layer before the model makes the final decision. The majority of works fall into the first group. For example, [22] use concepts from WordNet and NELL, and weighted average vectors of the retrieved concepts to calculate a new LSTM state. [13] retrieve relevant concepts from external knowledge for each token,

and get an additional vector with a solution similar to the key-value memory network. We believe that this line might work well on a specific dataset; however, the model only learns overlapped knowledge between the task-specific data and the external knowledge base. Thus, the model may not be easily adapted to another task/dataset where the overlapped is different from the current one.

Our work relates to the field of model pretraining in NLP and computer vision fields [11]. In the NLP community, works on model pretraining can be divided into unstructured text-based and structured knowledge-based ones. Both word embedding learning algorithms [17] and contextual embedding learning algorithms [5] belong to the text-based direction. Compared with these methods, which aim to learn a representation for a continuous sequence of words, our goal is to model the concept relatedness with graph structure in the knowledge base. Previous works on knowledge-based pretraining are typically validated on knowledge base completion or link prediction task [3]. We believe that combining both structured knowledge graphs and unstructured texts to do model pretraining is very attractive, and we leave this for future work.

8 Conclusion

We work on commonsense based question answering tasks. We present a simple and effective way to pre-train models to measure relations between concepts. Each concept is represented based on its internal information (i.e., the words it contains) and external context (i.e., neighbors in the knowledge graph). We use ConceptNet as the external commonsense knowledge base, and apply the pre-trained model on three question answering tasks (ARC, SemEval and OpenBook QA). Results show that the pre-trained models are complementary to standard document-based neural network approaches and could make further improvement through model combination.

Acknowledge

This work is supported by the National Key R&D Program of China (2018YFB1004404), Key R&D Program of Guangdong Province (2018B010107005), National Natural Science Foundation of China (U1711262, U1401256,U1501252,U1611264, U1711261,61673403,U1611262). The corresponding author is Jian Yin.

References

1. Annervaz, K., Chowdhury, S.B.R., Dukkipati, A.: Learning beyond datasets: Knowledge graph augmented neural networks for natural language processing. arXiv preprint arXiv:1802.05930 (2018)
2. Boratko, M., Padigela, H., Mikkilineni, D., Yuvraj, P., Das, R., McCallum, A., Chang, M., Fokoue-Nkoutche, A., Kapanipathi, P., Mattei, N., et al.: A systematic classification of knowledge, reasoning, and context within the arc dataset. arXiv preprint arXiv:1806.00358 (2018)

3. Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. In: In NIPS. pp. 2787–2795 (2013)
4. Clark, P., Cowhey, I., Etzioni, O., Khot, T., Sabharwal, A., Schoenick, C., Tafjord, O.: Think you have solved question answering? try arc, the ai2 reasoning challenge. arXiv preprint arXiv:1803.05457 (2018)
5. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
6. Hirsch, E.D.: Reading comprehension requires knowledge—of words and the world. American Educator **27**(1), 10–13 (2003)
7. Jia, R., Liang, P.: Adversarial examples for evaluating reading comprehension systems. arXiv preprint arXiv:1707.07328 (2017)
8. Lai, G., Xie, Q., Liu, H., Yang, Y., Hovy, E.: Race: Large-scale reading comprehension dataset from examinations. arXiv preprint arXiv:1704.04683 (2017)
9. Levesque, H.J., Davis, E., Morgenstern, L.: The winograd schema challenge. In: Aaai spring symposium: Logical formalizations of commonsense reasoning. vol. 46, p. 47 (2011)
10. Liang, P.: Learning executable semantic parsers for natural language understanding. Communications of the ACM **59**(9), 68–76 (2016)
11. Mahajan, D., Girshick, R., Ramanathan, V., He, K., Paluri, M., Li, Y., Bharambe, A., van der Maaten, L.: Exploring the limits of weakly supervised pretraining. arXiv preprint arXiv:1805.00932 (2018)
12. Mihaylov, T., Clark, P., Khot, T., Sabharwal, A.: Can a suit of armor conduct electricity? a new dataset for open book question answering. arXiv preprint arXiv:1809.02789 (2018)
13. Mihaylov, T., Frank, A.: Knowledgeable reader: Enhancing cloze-style reading comprehension with external commonsense knowledge. arXiv preprint arXiv:1805.07858 (2018)
14. Miller, A.H., Fisch, A., Dodge, J., Karimi, A., Bordes, A., Weston, J.: Key-value memory networks for directly reading documents. CoRR **abs/1606.03126** (2016), http://arxiv.org/abs/1606.03126
15. Mitchell, J., Lapata, M.: Composition in distributional models of semantics. Cognitive science **34**(8), 1388–1429 (2010)
16. Ostermann, S., Roth, M., Modi, A., Thater, S., Pinkal, M.: Semeval-2018 task 11: Machine comprehension using commonsense knowledge. In: Proceedings of The 12th International Workshop on Semantic Evaluation. pp. 747–757 (2018)
17. Pennington, J., Socher, R., Manning, C.: Glove: Global vectors for word representation. In: In EMNLP. pp. 1532–1543 (2014)
18. Rajpurkar, P., Zhang, J., Lopyrev, K., Liang, P.: Squad: 100,000+ questions for machine comprehension of text. arXiv preprint arXiv:1606.05250 (2016)
19. Scarselli, F., Gori, M., Tsoi, A.C., Hagenbuchner, M., Monfardini, G.: The graph neural network model. IEEE Transactions on Neural Networks **20**(1), 61–80 (2009)
20. Speer, R., Havasi, C.: Representing general relational knowledge in conceptnet 5. In: LREC. pp. 3679–3686 (2012)
21. Tandon, N., de Melo, G., Weikum, G.: Webchild 2.0: fine-grained commonsense knowledge distillation. Proceedings of ACL 2017, System Demonstrations pp. 115–120 (2017)
22. Yang, B., Mitchell, T.: Leveraging knowledge bases in lstms for improving machine reading. In: In ACL. pp. 1436–1446 (2017)