Fusing Context Into Knowledge Graph for Commonsense Reasoning

Yichong Xu*, **Chenguang Zhu***, **Ruochen Xu, Yang Liu, Michael Zeng, Xuedong Huang**
Microsoft Cognitive Services Research Group
{yicxu,chezhu,ruox,yaliu10,nzeng,xdh}@microsoft.com

Abstract

Commonsense reasoning requires a model to make presumptions about world events via language understanding. Many methods couple pre-trained language models with knowledge graphs in order to combine the merits in language modeling and entity-based relational learning. However, although a knowledge graph contains rich structural information, it lacks the context to provide a more precise understanding of the concepts and relations. This creates a gap when fusing knowledge graphs into language modeling, especially in the scenario of insufficient paired text-knowledge data. In this paper, we propose to utilize external entity description to provide contextual information for graph entities. For the CommonsenseQA task, our model first extracts concepts from the question and choice, and then finds a related triple between these concepts. Next, it retrieves the descriptions of these concepts from Wiktionary and feed them as additional input to a pre-trained language model, together with the triple. The resulting model can attain much more effective commonsense reasoning capability, achieving state-of-the-art results in the CommonsenseQA dataset with an accuracy of 80.7% (single model) and 83.3% (ensemble model) on the official leaderboard.

1 Introduction

One critical aspect of human intelligence is the ability to reason over everyday matters based on observation and knowledge. This capability is usually shared by most people as a main foundation for communication and interaction with the world. Therefore, commonsense reasoning has emerged as an important task in natural language understanding. Various datasets and models have been proposed in this area (Ma et al., 2019; Talmor et al., 2018; Wang et al., 2020; Lv et al., 2020).

While massive pre-trained models such as BERT (Devlin et al., 2018) and RoBERTa (Liu et al., 2019) are effective in language understanding, they lack modules to explicitly handle knowledge and commonsense. Also, text is much less efficient in representing commonsense compared with structured data. For example, to understand that the painting Mona Lisa is in Louvre, it requires multiple sentences containing this fact for the language model to implicitly encode this information, whereas an edge with relation "LocatedAt" between two entity nodes "Mona Lisa" and "Louvre" can exactly represent the same information. Therefore, there have been multiple methods coupling language models with various forms of knowledge graphs for commonsense reasoning, including knowledge bases (Tandon et al., 2017; Sap et al., 2019), relational paths (Lin et al., 2019), graph relation network (Feng et al., 2020) and heterogeneous graph (Lv et al., 2020). These methods combine the merits of language modeling and structural knowledge information, and improve the performance on commonsense reasoning.

However, there is still a non-negligible gap between the performance of these models and humans. One reason is that although a knowledge graph can encode topological information between the concepts, it lacks rich context information. For instance, for the entity node "Mona Lisa", the graph depicts its relations to multiple other entities. But given this neighborhood information, it is still hard to infer that it is a painting. On the other hand, we can retrieve the precise definition of "Mona Lisa" from external sources, e.g. Wiktionary: *A painting by Leonardo da Vinci, widely considered as the most famous painting in history*. Therefore, to produce a representation of structured data that can be seamlessly integrated into language models, we need to provide a panoramic view of each concept in the knowledge graph, including its neighboring

[*] Equal contribution

concepts, relations to them and a definitive description of it.

Thus, we propose the DEKCOR model, i.e. DEscriptive Knowledge for COmmonsense Reasoning. Given a commonsense question and a choice, we first extract the contained concepts. Then, we extract the edge between the question concept and the choice concept in ConceptNet (Liu and Singh, 2004). If such an edge does not exist, we compute a relevance score for each triple (node-edge-node) containing the choice concept, and select the one with the highest score. Next, we retrieve the definition of these concepts from Wiktionary via multiple criteria of text matching. Finally, we feed the question, choice, selected triple and definitions into the language model Albert (Lan et al., 2019), and the relevance score is generated by the appended attention layer and softmax layer.

We evaluate our model on CommonsenseQA dataset and DEKCOR outperforms the previous state-of-the-art result by 1.2% (single model) and 3.8% (ensemble model) in the test set, becoming the first model to surpass the accuracy of 80%. We further conduct ablation study to demonstrate the effectiveness of fusing context into knowledge graph.

2 Related work

Leveraging external knowledge sources to answer commonsense questions has been investigated with different approaches. Min et al. (2019) addresses open-domain QA by retrieving from a passage graph, where vertices are passages and edges represent relationships derived from external knowledge bases and co-occurrence. Sap et al. (2019) introduces the ATOMIC graph with 877k textual descriptions of inferential knowledge (e.g. if-then relation) to answer causal questions. Lin et al. (2019) projects questions and choices to the knowledge-based symbolic space as a schema graph. It then utilizes path-based LSTM to give scores. Feng et al. (2020) adopts the multi-hop graph relation network (MHGRN) to perform reasoning which unifies path-based methods and graph neural networks to achieve better interpretability and scalability. Lv et al. (2020) proposes to extract evidence from both structured knowledge base such as ConceptNet and Wikipedia text and conduct graph-based representation and inference for commonsense reasoning. Wang et al. (2020) employs GPT-2 to generate knowledgeable paths between knowledge graph concepts, which can dynamically provide multi-hop relations between any pair of concepts.

There have been works utilizing knowledge descriptions for different tasks. Yu et al. (2020) uses description text from Wikipedia for knowledge-text co-pretraining. Xie et al. (2016) encodes the semantics of entity descriptions in knowledge graphs to improve the performance on knowledge graph completion and entity classification. Chen et al. (2018) co-trains the knowledge graph embeddings and entity description representation for cross-lingual entity alignment.

3 Method

3.1 Knowledge Retrieval

Problem formulation. Given a commonsense question Q and several answer choices $c_1, ..., c_n$, the task is to select the correct answer. In most cases, the question does not contain any mentions of the answer. Therefore, external knowledge source can be used to provide additional information. We adopt ConceptNet (Liu and Singh, 2004) as our knowledge graph $G = (V, E)$, which contains over 8 million entities as nodes and over 21 million relations as edges. In the following, we use triple to refer to two neighboring nodes and the edge connecting them, i.e. $(u \in V, p = (u, v) \in E, v \in V)$.

For each question and answer we get the corresponding concept in the knowledge graph provided by CommonsenseQA. Suppose the question entity is $e_q \in V$ and the choice entity is $e_c \in V$. In order to conduct knowledge reasoning, we employ the KCR method (Knowledge Chosen by Relations)[1]. If there is a direct edge r from e_q to e_c in G, we choose this triple (e_q, r, e_c). Otherwise, we retrieve all the N triples containing e_c. Each triple j is assigned a score s_j which is the product of its triple weight w_j (provided by ConceptNet) and relation type weight t_{r_j}:

$$s_j = w_j \cdot t_{r_j} \tag{1}$$

$$t_{r_j} = \frac{N}{N_{r_j}} \tag{2}$$

Here, r_j is the relation type of the triple j, and N_{r_j} is the number of triples among the retrieved triples that have the relation type r_j. Finally, the triple with the highest weight is chosen.

[1] https://github.com/jessionlin/csqa/blob/master/Model_details.md

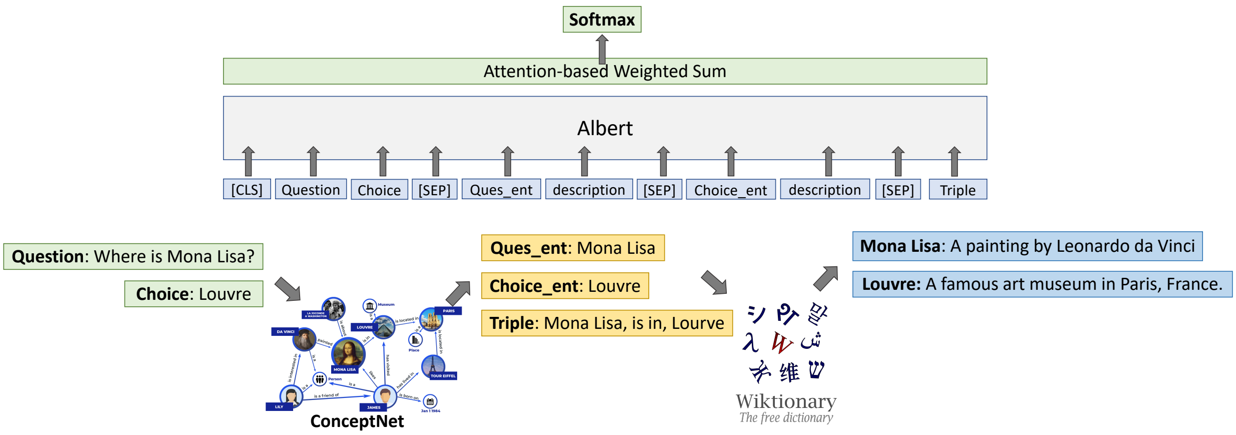

Figure 1: The input to Albert includes the question, choice, entity names, description text and triple. An attention-based weighted sum and a softmax layer processes the output from Albert to produce the prediction.

3.2 Contextual information

The retrieved entities and relations from the knowledge graph are described by their surface form. Without additional context, it is hard to for the language model to understand its exact meaning, especially for proper nouns.

Therefore, we leverage large-scale online dictionaries to provide definitions as context. We use the dump of Wiktionary[2] which includes definitions of 999,614 concepts. For every concept, we choose its first definition entry in Wiktionary as the description. For every question/choice concept, we find its closest match in Wiktionary by using the following forms in order: i) original form; ii) lemma form by Spacy (Honnibal and Montani, 2017); iii) base word (last word). For example, the concept "taking notes" does not appear in its original form in Wiktionary , but its lemma form "take notes" is in Wiktionary and we get its description text: *To make a record of what one hears or observes for future reference*. We find descriptions of all entities in our experiments. The descriptions of the question and choice concept are denoted by d_q and d_c, respectively.

Finally, we feed the question, answer, descriptions and triple (from Section 3.1) into the Albert (Lan et al., 2019) encoder in the following format: [CLS] Q c_i [SEP] e_q: d_q [SEP] e_c: d_c [SEP] triple.

3.3 Reasoning

On top of the output from Albert, we leverage an attention-based weighted sum and a softmax layer to generate the relevance score for the question-choice pair.

Table 1: Example question and answer choices in CommonsenseQA. The correct choice is in bold.

Q: Where can I stand on a river to see water falling without getting wet?
(A) waterfall, **(B) bridge**, (C) valley, (D) stream, (E) bottom

In detail, suppose the output representations of Albert is $(\boldsymbol{x}_0, ..., \boldsymbol{x}_m)$, where $\boldsymbol{x}_i \in R^d$. We compute a weighted sum of these embeddings based on attention:

$$q_i = \boldsymbol{u}^T \boldsymbol{W} \boldsymbol{x}_i \tag{3}$$

$$\alpha_i = \text{softmax}(q_i) \tag{4}$$

$$\boldsymbol{v} = \sum_{i=0}^{m} \alpha_i \boldsymbol{x}_i, \tag{5}$$

where $\boldsymbol{u}$ is a parameter vector and $\boldsymbol{W} \in R^{d\times d}$ is a parameter matrix.

The relevance score of the question and the j-th choice is then $s_j = \text{softmax}(\boldsymbol{v}^T \boldsymbol{b})$, where $\boldsymbol{b} \in R^d$ is a parameter vector and the softmax is computed over all choices for the cross entropy loss function.

The architecture of DEKCOR model and the input construction is shown in Fig. 1.

4 Experiments

4.1 Datasets

We evaluate our model on the benchmark dataset for commonsense reasoning: CommonsenseQA (Talmor et al., 2018). This dataset contains 12,102 examples, which include 9,741 for training, 1,221

[2] https://www.wiktionary.org/

Table 2: Accuracy on the test set of CommonsenseQA.

Methods	Single	Ensemble
BERT+OMCS	62.5	-
RoBERTa (Liu et al., 2019)	72.1	72.5
RoBERTa+FreeLB (Zhu et al., 2019)	72.2	73.1
RoBERTa+HyKAS (Ma et al., 2019)	73.2	-
XLNet+DREAM	-	73.3
RoBERTa+KE	73.3	-
RoBERTa+KEDGN	-	74.4
XLNet+GraphReason (Lv et al., 2020)	75.3	-
Albert (Lan et al., 2019)	-	76.5
RoBERTa+MHGRN (Feng et al., 2020)	75.4	76.5
Albert+PG-Full (Wang et al., 2020)	75.6	78.2
T5 (Raffel et al., 2019)	78.1	-
Albert+KRD	78.4	-
UnifiedQA (Khashabi et al., 2020)	79.1	-
Albert+KCR	79.5	-
DEKCOR (ours)	**80.7**	**83.3**

Table 3: Ablation results on the dev set of CommonsenseQA.

Methods	Accuracy
DEKCOR	84.7
– Description	82.0
– Triple	80.1

for development and 1,140 for test. Each example consists of a question and up to five answer choices (Table 1). The name of the concept in the question is also given, which corresponds to an entity in ConceptNet.

4.2 Baselines

We compare our models with state-of-the-art baselines on CommonsenseQA. All baselines employ pre-trained models including RoBERTa (Liu et al., 2019), XLNet (Yang et al., 2019), Albert (Lan et al., 2019) and T5 (Raffel et al., 2019). Some baselines employ additional modules to process knowledge information. XLNet+GraphReason (Lv et al., 2020) retrieves knowledge from both structured knowledge base (i.e. ConceptNet) and Wikipedia plain text. Albert+PG-FULL (Wang et al., 2020) fine-tunes GPT-2 on ConceptNet to generate knowledgeable paths between knowledge graph concepts. RoBERTa+MHGRN (Feng et al., 2020) adopts the multi-hop graph relation network to perform reasoning on ConceptNet with both path-based methods and graph neural networks. RoBERTa+HyKAS (Ma et al., 2019) employs an option comparison network based on RoBERTa to consume triples from ConceptNet. Albert+KRD retrieves top k commonsense knowledge from Open Mind Common Sense for the question-choice pair and then uses a self-attention module to compute a weighted sum of these triple representations.

4.3 Implementation Details

We use the AdamW (Loshchilov and Hutter, 2017) optimizer with a learning rate of 2e-5. The batch size is 8. We limit the maximum length of input sequence to 192 tokens. The model is trained for 10 epochs. We use the Huggingface (Wolf et al., 2019) implementation for the Albert model.

For the ensemble model, we train 7 single models with different initialization random seeds. The output of the ensemble model is the majority of choices selected by these single models.

4.4 Results

Table 2 shows the accuracy on the official test set of CommonsenseQA. For fair comparison, we categorize the results into single models and ensemble models. As shown, our proposed DEKCOR outperforms the previous state-of-the-art result by 1.2% (single model) and 3.8% (ensemble model). This demonstrates the effectiveness of the usage of knowledge description to provide context.

Furthermore, we notice two trends based on the results. First, the underlying pre-trained language model is important in commonsense reasoning quality. In general, we observe this order of accuracy among these language models: BERT<RoBERTa<XLNet<Albert<T5. Second, the additional knowledge module is critical to provide external information for reasoning. For example, RoBERTa+KEDGN outperforms the vanilla RoBERTa by 1.9% in accuracy, and our model outperforms the vanilla Albert model by 6.8% in accuracy.

Ablation study. Table 3 shows that the usage of concept descriptions from Wiktionary and triple from ConceptNet can help improve the accuracy of DEKCOR on the dev set of CommonsenseQA by 2.7% and 4.6% respectively. This demonstrates that additional context information can help with fusing knowledge graph into language modeling for commonsense reasoning.

5 Conclusions

In this paper, we propose to fuse context information into knowledge graph for commonsense reasoning. As a knowledge graph often lacks description for the contained entities and edges, we leverage Wiktionary to provide definitive text for each question/choice entity. This description is combined with entity names and sent into a pretrained language model to produce predictions. The resulting DEKCOR model achieves state-of-the-art result on the benchmark dataset CommonsenseQA.

References

Muhao Chen, Yingtao Tian, Kai-Wei Chang, Steven Skiena, and Carlo Zaniolo. 2018. Co-training embeddings of knowledge graphs and entity descriptions for cross-lingual entity alignment. *arXiv preprint arXiv:1806.06478*.

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

Yanlin Feng, Xinyue Chen, Bill Yuchen Lin, Peifeng Wang, Jun Yan, and Xiang Ren. 2020. Scalable multi-hop relational reasoning for knowledge-aware question answering. *arXiv preprint arXiv:2005.00646*.

Matthew Honnibal and Ines Montani. 2017. spacy 2: Natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing. *To appear*, 7(1).

Daniel Khashabi, Tushar Khot, Ashish Sabharwal, Oyvind Tafjord, Peter Clark, and Hannaneh Hajishirzi. 2020. Unifiedqa: Crossing format boundaries with a single qa system. *arXiv preprint arXiv:2005.00700*.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*.

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

Hugo Liu and Push Singh. 2004. Conceptnet—a practical commonsense reasoning tool-kit. *BT technology journal*, 22(4):211–226.

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

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Shangwen Lv, Daya Guo, Jingjing Xu, Duyu Tang, Nan Duan, Ming Gong, Linjun Shou, Daxin Jiang, Guihong Cao, and Songlin Hu. 2020. Graph-based reasoning over heterogeneous external knowledge for commonsense question answering.

Kaixin Ma, Jonathan Francis, Quanyang Lu, Eric Nyberg, and Alessandro Oltramari. 2019. Towards generalizable neuro-symbolic systems for commonsense question answering. *arXiv preprint arXiv:1910.14087*.

Sewon Min, Danqi Chen, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2019. Knowledge guided text retrieval and reading for open domain question answering. *arXiv preprint arXiv:1911.03868*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*.

Maarten Sap, Ronan Le Bras, Emily Allaway, Chandra Bhagavatula, Nicholas Lourie, Hannah Rashkin, Brendan Roof, Noah A Smith, and Yejin Choi. 2019. Atomic: An atlas of machine commonsense for if-then reasoning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3027–3035.

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

Niket Tandon, Gerard De Melo, and Gerhard Weikum. 2017. Webchild 2.0: Fine-grained commonsense knowledge distillation. In *Proceedings of ACL 2017, System Demonstrations*, pages 115–120.

Peifeng Wang, Nanyun Peng, Pedro Szekely, and Xiang Ren. 2020. Connecting the dots: A knowledgeable path generator for commonsense question answering. *arXiv preprint arXiv:2005.00691*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, pages arXiv–1910.

Ruobing Xie, Zhiyuan Liu, Jia Jia, Huanbo Luan, and Maosong Sun. 2016. Representation learning of knowledge graphs with entity descriptions.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. pages 5753–5763.

Donghan Yu, Chenguang Zhu, Yiming Yang, and Michael Zeng. 2020. Jaket: Joint pre-training of knowledge graph and language understanding. *arXiv preprint arXiv:2010.00796*.

Chen Zhu, Yu Cheng, Zhe Gan, Siqi Sun, Tom Goldstein, and Jingjing Liu. 2019. Freelb: Enhanced adversarial training for natural language understanding.