Graph-Based Reasoning over Heterogeneous External Knowledge for Commonsense Question Answering

Shangwen Lv\(^1\), Daya Guo\(^2\), Jingjing Xu\(^3\), Duyu Tang\(^4\), Nan Duan\(^4\), Ming Gong\(^4\), Linjun Shou\(^4\), Daxin Jiang\(^4\), Guihong Cao\(^4\), Songlin Hu\(^4\)

\(^1\)Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
\(^2\)Sun Yat-sen University
\(^3\)Peking University
\(^4\)Microsoft Corporation

Abstract

Commonsense question answering aims to answer questions which require background knowledge that is not explicitly expressed in the question. The key challenge is how to obtain evidence from external knowledge and make predictions based on the evidence. Recent works either learn to generate evidence from human-annotated evidence which is expensive to collect, or extract evidence from either structured or unstructured knowledge bases which fails to take advantages of both sources. In this work, we propose to automatically extract evidence from heterogeneous knowledge sources, and answer questions based on the extracted evidence. Specifically, we extract evidence from both structured knowledge base (i.e. ConceptNet) and Wikipedia plain texts. We construct graphs for both sources to obtain the relational structures of evidence. Based on these graphs, we propose a graph-based approach consisting of a graph-based contextual word representation learning module and a graph-based inference module. The first module utilizes graph structural information to re-define the distance between words for learning better contextual word representations. The second module adopts graph convolutional network to encode neighbor information into the representations of nodes, and aggregates evidence with graph attention mechanism for predicting the final answer. Experimental results on CommonsenseQA dataset illustrate that our graph-based approach over both knowledge sources brings improvement over strong baselines. Our approach achieves the state-of-the-art accuracy (75.3\%) on the CommonsenseQA leaderboard.

Introduction

Reasoning is an important and challenging task in artificial intelligence and natural language processing, which is “the process of drawing conclusions from the principles and evidence” (Wason and Johnson-Laird 1972). The “evidence” is the fuel and the “principle” is the machine that operates on the fuel to make predictions. The majority of studies typically only take the current datapoint as the input, in which case the important “evidence” of the datapoint from background knowledge is ignored.

In this work, we study commonsense question answering, a challenging task which requires machines to collect background knowledge and reason over the knowledge to answer questions. For example, an influential dataset CommonsenseQA (Talmor et al. 2019) is built in a way that the answer choices share the same relation with the concept in the question while annotators are asked to use their background knowledge to create questions so that only one choice is the correct answer. Figure 1 shows an example which requires multiple external knowledge sources to make the correct predictions. The structured evidence from ConceptNet can help pick up the two choices (A, C) and Wikipedia evidence helps pick up choices (C, E). Combining both evidence will derive the right answer C. Words in blue are the concepts in the question. Words in green are the evidence from ConceptNet. Words in red are the choices picked up by evidence.

- **Evidence from ConceptNet**
  - A. cry
  - B. hear sounds
  - C. singing
  - D. arthritis
  - E. making music

- **Evidence from Wikipedia**
  - A. cry
    - What can yearn, cry without tears?
    - What is to cry and to weep?
  - C. singing
    - She also performed them, playing guitar and singing.
  - E. making music
    - I like making music and playing guitar with other people.

Figure 1: An example from the CommonsenseQA dataset which requires multiple external knowledge to make the correct prediction. ConceptNet evidence helps pick up choices (A, C) and Wikipedia evidence helps pick up choices (C, E). Combining both evidence will derive the right answer C. Words in blue are the concepts in the question. Words in green are the evidence from ConceptNet. Words in red are the choices picked up by evidence.
extracting evidence and reasoning over evidence. Typically, they either generate evidence from human-annotated evidence (Rajani et al. 2019) or extract evidence from a homogeneous knowledge source like structured knowledge ConceptNet (Bill Yuchen Lin 2019), Bauer, Wang, and Bansal 2018, Mihaylov and Frank 2018 or Wikipedia plain texts (Ryu, Jang, and Kim 2014) (Yang, Yih, and Meek 2015) (Chen et al. 2017), but they fail to take advantages of both knowledge sources. Structured knowledge sources contain valuable structural relations between concepts, which are beneficial for reasoning. However, they suffer from low coverage. Plain texts can provide abundant and high-coverage evidence, which is complementary to the structured knowledge.

In this work, we study commonsense question answering by using automatically collected evidence from heterogeneous external knowledge. Our approach consists of two parts: knowledge extraction and graph-based reasoning. In the knowledge extraction part, we automatically extract graph paths from ConceptNet and sentences from Wikipedia. To better use the relational structure of the evidence, we construct graphs for both sources, including extracted graph paths from ConceptNet and triples derived from Wikipedia sentences by Semantic Role Labeling (SRL). In the graph-based reasoning part, we propose a graph-based approach to make better use of the graph information. We contribute by developing two graph-based modules, including (1) a graph-based contextual word representation learning module, which utilizes graph structural information to re-define the distance between words for learning better contextual word representations, and (2) a graph-based inference module, which first adopts Graph Convolutional Network (Kipf and Welling 2016) to encode neighbor information into the representations of nodes, followed by a graph attention mechanism for evidence aggregation.

We conduct experiments on the CommonsenseQA benchmark dataset. Results show that both the graph-based contextual representation learning module and the graph-based inference module boost the performance. We also demonstrate that incorporating both knowledge sources can bring further improvements. Our approach achieves the state-of-the-art accuracy (75.3%) on the CommonsenseQA leaderboard.

Our contributions of this paper can be summarized as follows:

• We introduce a graph-based approach to leverage evidence from heterogeneous knowledge sources for commonsense question answering.

• We propose a graph-based contextual representation learning module and a graph-based inference module to make better use of the graph information for commonsense question answering.

• Results show that our model achieves a new state-of-the-art performance on the CommonsenseQA leaderboard.

Task Definition and Dataset

This paper utilizes CommonsenseQA (Talmor et al. 2019), an influential dataset for commonsense question answering task for experiments. Formally, given a natural language question \( Q \) containing \( m \) tokens \( \{q_1, q_2, \cdots, q_m\} \), and 5 choices \( \{a_1, a_2, \cdots, a_5\} \), the target is to distinguish the right answer from the wrong ones and accuracy is adopted as the metric. The choices share the same relation in ConceptNet with the concept in the question. Annotators are required to utilize their background knowledge to write questions in which only one of them is correct, thus making the task more challenging. It has a broad coverage of many commonsense relations like temporal, causal, physical, spatial, etc. The lack of evidence requires the model to have strong commonsense knowledge extraction and reasoning ability to get the right results. In addition, the authors build a leaderboard to show the results on the test dataset\(^1\).
Knowledge Extraction

In this section, we provide the methods to extract evidence from ConceptNet and Wikipedia given the question and choices. Furthermore, we describe the details of constructing graphs for both sources.

Knowledge Extraction from ConceptNet

ConceptNet is a large-scale commonsense knowledge base, containing millions of nodes and relations. The triple in ConceptNet contains four parts: two nodes, one relation, and a relation weight. For each question and choice, we first identify their entities in the given ConceptNet graph. Then we search for the paths (less than 3 hops) from question entities to choice entities and merge the covered triples into a graph where nodes are triples and edges are the relation between triples. If two triples $s_i$, $s_j$ contain the same entity, we will add an edge from the previous triple $s_i$ to the next triple $s_j$. In order to obtain contextual word representations for ConceptNet nodes, we transfer the triple into a natural language sequence according to the relation template in ConceptNet. We denote the graph as Concept-Graph.

Know edge

We extract 107M sentences from Wikipedia by Spacy and adopt Elastic Search tools to index the Wikipedia sentences. We first remove stopwords in the given question and choices then concatenate the words as queries to search from the Elastic Search engine. The engine ranks the matching scores between queries and all the Wikipedia sentences. We select top $K$ sentences as the Wikipedia evidence. Here we adopt $K=10$ in experiments.

To discover the structure information in Wikipedia evidence, we construct a graph for Wikipedia evidence. We utilize Semantic RoleLabeling (SRL) to extract arguments (subjective, objective) for each predicate in one sentence. Both arguments and predicates are the nodes in the graph. The relations between predicates and arguments are the edges in the graph. In order to enhance the connectivity of the graph, we remove stopwords and add an edge from node $a$ to node $b$ according to the following rules: (1) Node $a$ is contained in node $b$ and the number of words in $a$ is more than 3; (2) Node $a$ and node $b$ only have one different word and the numbers of words in $a$ and $b$ are both more than 3. We denote the Wikipedia graph as Wiki-Graph.

Graph-Based Reasoning

In this section, we present the model architecture of graph-based reasoning over the extracted evidence, shown in Figure 3. Our graph-based model consists of two modules: a graph-based contextual representation learning module and a graph-based inference module. The first module learns better contextual word representations by using graph information to re-define the distance between words. The second module gets node representations via Graph Convolutional

Network (Kipf and Welling 2016) by using neighbor information and aggregates graph representations to make final predictions.

Graph-Based Contextual Representation Learning Module

It is well accepted that pre-trained models have a strong text understanding ability and have achieved state-of-the-art results on a variety of natural language processing tasks. We use XLNet (Yang et al. 2019) as the backbone here, which is a successful pre-trained model with the advantage of capturing long-distance dependency. A simple way to get the representation of each word is to concatenate all the evidence as a single sequence and feed the raw input into XLNet. However, this would assign a long distance for the words mentioned in different evidence sentences, even though they are semantically related. Therefore, we use the graph structure to re-define the relative position between evidence words. In this way, semantically related words will have shorter relative position and the internal relational structures in evidence are used to obtain better contextual word representations.

Specifically, we develop an efficient way of utilizing topology sort algorithm to re-order the input evidence according to the constructed graphs. For Wikipedia sentences, we construct a sentence graph. The evidence sentences $S$ are nodes in the graph. For two sentences $s_i$ and $s_j$, if there is an edge $(p, q)$ in Wiki-Graph where $p, q$ are in $s_i$ and $s_j$ respectively, there will be an edge $(s_i, s_j)$ in the sentence graph. We can get a sorted evidence sequence $S'$ by the method in Algorithm 1. For structured knowledge, ConceptNet triples are

---

3Wikipedia version enwiki-20190301
4https://spacy.io/
5https://www.elastic.co/
**Algorithm 1** Topology Sort Algorithm.

Require: A sequence of nodes \( S = \{ s_1, s_2, \cdots, s_n \} \); A set of relations \( R = \{ r_1, r_2, \cdots, r_m \} \).

1: function DFS(node, visited, sorted_sequence)
2:   for each child \( s_c \) in node’s children do
3:     if \( s_c \) has no incident edges and \( \text{visited}[s_c] == 0 \) then
4:       \( \text{visited}[s_c] = 1 \)
5:       DFS(s_c, visited)
6:   end if
7: end for
8: sorted_sequence.append(0, s_c)
9: end function
10: sorted_sequence = []
11: visited = [0 for i in range(n)]
12: S, R = to_acyclic_graph(S, R)
13: for each node \( s_i \) in \( S \) do
14:   if \( s_i \) has no incident edges and \( \text{visited}[i] == 0 \) then
15:     \( \text{visited}[i] = 1 \)
16:     for each child \( s_c \) in \( s_i \)’s children do
17:       DFS(s_c, visited, sorted_sequence)
18: end for
19: end if
20: end for
21: return sorted_sequence

The \( i \)-th node representation \( h^0_i \) is obtained by averaging hidden states of the corresponding evidence in the output of XLNet and reducing dimension via a non-linear transformation.

\[
h^0_i = \sigma(W \sum_{w_j \in s_i} \frac{1}{|s_i|} h_{w_j})
\]

where \( s_i = \{w_0, \cdots, w_l\} \) is the corresponding evidence to the \( i \)-th node, \( h_{w_j} \) is the contextual token representation of XLNet for the token \( w_j \), \( W \in \mathbb{R}^{d \times k} \) is to reduce high dimension \( d \) into low dimension \( k \), and \( \sigma \) is an activation function.

In order to reason over the graph, we propagate information across evidence via two steps: aggregation and combination [Hamilton, Ying, and Leskovec 2017]. The first step aggregates information from neighbors of each node. The aggregated information \( z^l_i \) for \( i \)-th node can be formulated as Equation 2, where \( N_i \) is the neighbors of \( i \)-th node and \( h^l_i \) is the \( j \)-th node representation at the layer \( l \). The representation \( z^l_i \) contains neighbors information for \( i \)-th node at the layer \( l \), and we can combine it with the transformed \( i \)-th node representation to get the updated node representation \( h^{l+1}_i \).

\[
z^l_i = \sum_{j \in N_i} \frac{1}{|N_i|} V^l h^l_j
\]

\[
h^{l+1}_i = \sigma(W^l h^l_i + z^l_i)
\]

We utilize graph attention to aggregate graph-level representations to make the prediction. The graph representation is computed the same as the multiplicative attention [Luong, Pham, and Manning 2015], where \( h^c_i \) is the \( i \)-th node representation at the last layer, \( h^c \) is the representation of the last token in XLNet and can be regarded as the input representation, \( \alpha_i \) is the importance of the \( i \)-th node, and \( h^g \) is the graph representation.

\[
\alpha_i = \frac{h^c \sigma(W_1 h^l_i)}{\sum_{j \in N} h^c \sigma(W_1 h^l_j)}
\]

\[
h^g = \sum_{j \in N} \alpha_j h^l_j
\]

We concatenate the input representation \( h^g \) with the graph representation \( h^g \) as the input of a Multi-Layer Perceptron (MLP) to compute the confidence score \( \text{score}(q, a) \). The probability of the answer candidate \( a \) to the question \( q \) can be computed as follows, where \( A \) is the set of candidate answers.

\[
p(q,a) = \frac{e^{\text{score}(q,a)}}{\sum_{a' \in A} e^{\text{score}(q,a')}}
\]

Finally, we select the answer with the highest confidence score as the predicted answer.

**Experiments**

In this section, we conduct experiments to prove the effectiveness of our proposed approach. To dig into our approach,
we perform ablation studies to explore the different effects of heterogeneous knowledge sources and graph-based reasoning models. We study a case to show how our model can utilize the extracted evidence to get the right answer. We also show some error cases to point directions to improve our model.

**Experiment Settings**

The CommonsenseQA (Talmor et al. 2019) dataset contains 12,102 examples, including 9,741 for training, 1,221 for development, and 1,140 for test. We select the best model on development dataset and submit the predicted answers on test dataset to the leaderboard.

We select XLNet large cased (Yang et al. 2019) as the pre-trained model. We concatenate “The answer is” before each choice to change each choice to a sentence. The input format for each choice is “<evidence> <sep> question <sep> The answer is <choice> <cls>”. Totally, we get 5 confidences scores for all the choices then we adopt the softmax function to calculate the loss between the predictions and the ground truth. We adopt cross-entropy loss as our loss function. In our best model on the development dataset, we set the batch size to 4 and learning rate to 5e-6. We set max length of input to 256. We use Adam (Kingma and Ba 2014) with \( \beta_1 = 0.9, \beta_2 = 0.999 \) for optimization. We set GCN layer to 1. We train our model for 2,800 steps (about one epoch) and get the results 79.3% on development dataset and 75.3% on blind test dataset.

**Baselines**

For the compared methods, we select the submitted models from the leaderboard. We classify them into 4 groups. **Group 1**: models without descriptions or papers, **Group 2**: models without extracted knowledge, **Group 3**: models with extracted structured knowledge and **Group 4**: models with extracted unstructured knowledge.

- **Group 1**: models without description or papers. These models include SGN-lite, BECON (single), BECON (ensemble), CSR-KG and CSR-KG (AI2 IR).
- **Group 2**: models without extracted knowledge, including BERT-large (Devlin et al. 2019), XLNet-large (Yang et al. 2019) and RoBERTa (Liu et al. 2019). These models adopt pre-trained language models to finetune on the training data and make predictions directly on the test dataset without extracted knowledge.
- **Group 3**: models with extracted structured knowledge, including BERT + AMS (Ye et al. 2019) and BERT + CSPT. These models utilize structured knowledge ConceptNet to enhance the model to make predictions. BERT + AMS (Ye et al. 2019) constructs a commonsense-related multi-choice question answering dataset according to ConceptNet and pre-train on the generated dataset. BERT + CSPT first trains a generation model to generate synthetic data from ConceptNet, then finetunes RoBERTa on the synthetic data and Open Mind Common Sense (OMCS) corpus.
- **Group 4**: models with extracted unstructured knowledge, including CoS-E (Rajani et al. 2019), HyKAS, BERT + OMCS, AristoBERTv7, DREAM, RoBERT + KE, RoBERTa + IR and RoBERTa + CSPT. CoS-E (Rajani et al. 2019) constructs human-annotated evidence for each question and generates evidence for test data. HyKAS and BERT + OMCS models pre-train BERT whole word masking model on the OMCS corpus. AristoBERTv7 utilizes the information from machine reading comprehension data RACE (Lai et al. 2017) and extracts evidence from text sources such as Wikipedia, SimpleWikipedia, etc. DREAM adopts XLNet-large as the baseline and extracts evidence from Wikipedia. RoBERT + KE, RoBERTa + IR and RoBERTa + CSPT adopt RoBERTa as the baseline and utilize the evidence from Wikipedia, search engine and OMCS, respectively.

It should be noted that these methods either utilize evidence from structured or unstructured knowledge sources, failing to take advantages of both sources. RoBERT + CSPT adopts knowledge from ConceptNet and OMCS, but the model pre-trains on the sources without explicit reasoning over the evidence, which is different from our approach.

**Experiment Results and Analysis**

| Group | Model | Dev Acc | Test Acc |
|-------|-------|---------|----------|
| Group 1 | SGN-lite | - | 57.1 |
| | BECON (single) | - | 57.9 |
| | BECON (ensemble) | - | 59.6 |
| | CSR-KG | - | 61.8 |
| | CSR-KG (AI2 IR) | - | 65.3 |
| Group 2 | BERT-large | - | 56.7 |
| | XLNet-large | - | 62.9 |
| | RoBERTa (single) | 78.5 | 72.1 |
| | RoBERTa (ensemble) | - | 72.5 |
| Group 3 | BERT + AMS | - | 62.2 |
| | RoBERTa + CSPT | 76.2 | 69.6 |
| Group 4 | Cos-E | - | 58.2 |
| | KagNet | - | 58.9 |
| | BERT + OMCS | 68.8 | 62.5 |
| | HyKAS | - | 62.5 |
| | AristoBERTv7 | - | 64.6 |
| | DREAM | 73.0 | 66.9 |
| | RoBERT + KE | 77.5 | 68.4 |
| | RoBERTa + CSPT | 76.2 | 69.6 |
| | RoBERTa + IR | 78.9 | 72.1 |

| Our Model | 79.3 | 75.3 |

Table 1: Results on CommonsenseQA development and blind test dataset. All the results are obtained from the CommonsenseQA leaderboard.

The results on CommonsenseQA development dataset and blind test dataset are shown in Table 1. Our model achieves the best performance on both datasets. In the following comparisons we focus on the results on test dataset. Compared with the model in group 1, we can get more
than 10% absolute accuracy than these methods. Compared with models without extracted knowledge in group 2, our model also enjoys 2.8% absolute gain over the strong baseline RoBERTa (ensemble). XLNet-large is our baseline model and our approach can get 12.4% absolute improvement over the baseline and this approves the effectiveness of our approach. Compared to models with extracted structured knowledge in group 3, our model extracts graph paths from ConceptNet for graph-based reasoning rather than for pre-training, and we also extract evidence from Wikipedia plain texts, which brings 13.1% and 5.7% gains over BERT + AMS and ROBERTa + CHT respectively. Group 4 contains model which utilizes unstructured knowledge such as Wikipedia or OMCS, etc. Compared with these methods, we not only utilize Wikipedia to provide unstructured evidences but also construct graphs to get the structural information. We also utilize the evidence from structure knowledge base ConceptNet. Our model achieves 3.2% absolute improvement over the best model RoBERTa + IR in this part.

From the result analysis above, we can see that heterogeneous external knowledge sources and graph-based reasoning models help our model to obtain significant improvements and achieve a new state-of-the-art performance. In the next ablation section, we will dive into our model and see the influence of different knowledge sources and different components in our reasoning models.

**Ablation Study**

In this section, we perform ablation studies on the development dataset to dive into the effectiveness of different components in our model. We first explore the effect of different components in graph-based reasoning. Then we dive into the heterogeneous knowledge sources and see their effects.

In the graph-based reasoning part, we dive into the effect of topology sort algorithm for learning contextual word representations and graph inferences with GCN and graph attention. We select XLNet + Evidence as the baseline. In the baseline, we simply concatenate all the evidence into XLNet and adopt the contextual representation for prediction. By adding topology sort, we can obtain a 1.9% gain over the baseline. This proves that topology sort algorithm can fuse the graph structure information and change the relative position between words for better contextual word representation. The graph inference module brings 1.4% benefit, showing that GCN can obtain proper node representations and graph attention can aggregate both word and node representations to infer answers. Finally, we add topology sort, graph inference module together to get a 3.5% improvement, proving these models can be complementary and achieve better performance.

Then we perform ablations studies on knowledge sources to see the effectiveness of ConceptNet and Wikipedia sources. The results are shown in Table 3, “None” represents we only adopts the XLNet (Yang et al. 2019) large model as the baseline. When we add one knowledge source, the corresponding graph-based reasoning models are also added. From the results, we see that the structured knowledge ConceptNet can bring 6.4% absolute improvement and the Wikipedia source can bring 4.6% absolute improvement. This proves the benefits of ConceptNet or Wikipedia source. When combining ConceptNet and Wikipedia, we can enjoy a 9.4% absolute gain over the baseline. This proves that heterogeneous knowledge sources can achieve better performance than single one and different sources in our model and they are complementary to each other.

**Table 2:** Ablation studies on reasoning components in our model. E denotes evidence.

| Knowledge Sources | Dev Acc |
|-------------------|---------|
| XLNet + E         | 75.8    |
| XLNet + E + Topology Sort | 77.7    |
| XLNet + E + Graph Inference | 77.2    |
| XLNet + E + Topology Sort + Graph Inference | 79.3    |

**Table 3:** Ablation studies on heterogeneous knowledge sources. “None” represents we only use XLNet baseline and do not utilize knowledge sources.

| Knowledge Sources         | Dev Acc |
|---------------------------|---------|
| None                      | 68.9    |
| ConceptNet                | 75.3    |
| Wikipedia                 | 73.5    |
| ConceptNet + Wikipedia    | 79.3    |

**Case Study**

In this section, we select a case to show that our model can utilize the heterogeneous knowledge sources to answer questions. As shown in Figure 4, the question is “Animals who have hair and don’t lay eggs are what?” and the answer is “mammals”. The first three nodes are from ConceptNet evidence graph. We can see that “mammals is animals” and “mammals has hair” can provide information about the relation between “mammals” and two concepts “animals” and “hair”. More evidence is needed to show the relation between “lay eggs” and “mammals”. The last three nodes are from Wikipedia evidence graph and they can provide the information that “very few mammals lay eggs”. The examples also show that both sources are necessary to infer the right answer.

**Error Analysis**

We randomly select 50 error examples from the development dataset and the reasons are classified into three categories: the lack of evidence, similar evidence and dataset noise. There are 10 examples which are lack of evidence. For example, the first example in Figure 5 extracts no triples from ConceptNet and the evidence from Wikipedia does not contain enough information to get the right answer. This problem can be alleviated by utilizing more advanced extraction strategies and adding more knowledge.
ConceptNet

Wikipedia

Figure 4: An attention heat-map for the question “Animals who have hair and don’t lay eggs are what?” and the answer “mammals”. The nodes in ConceptNet are in natural language format and the template is: IsA (is a kind of), HasA (has).

Figure 5: Error cases of our model on the development dataset.

Related Work

Commonsense Reasoning Commonsense reasoning is a challenging direction since it requires reasoning over external knowledge beside the inputs to predict the right answer. Various downstream tasks have been released to address this problem like ATOMIC (Sap et al. 2019), Event2Mind (Rashkin et al. 2018), MScript 2.0 (Ostermann, Roth, and Pinkal 2019). Story Cloze Test (Mostafazadeh et al. 2016) aims to predict the right ending from a set of plausible ones given a series of stories. SWAG (Zellers et al. 2018) and HellaSWAG (Zellers et al. 2019) are two similar datasets trying to predict the next event given an initial event. The SWAG dataset has been well solved by pre-trained language models like BERT (Devlin et al. 2019). HellaSWAG is a more challenging dataset because the context and answer are longer and more difficult to understand.

Recently proposed CommonsenseQA (Talmor et al. 2019) dataset derived from ConceptNet (Speer, Chin, and Havasi 2017) and the choices have the same relation with the concept in the question. Recently, Rajani et al. (2019) explores adding human-written explanations to solve the problem. Bill Yuchen Lin (2019) extracts evidence from ConceptNet to study this problem. This paper focuses on automatically extracting evidence from heterogeneous external knowledge and reasoning over the extracted evidence to study this problem.

Knowledge Transfer in NLP Transfer learning has played a vital role in the NLP community. Pre-trained language models from large-scale unstructured data like ELMo (Peters et al. 2018), GPT (Radford et al. 2018), BERT (Devlin et al. 2019), XLNet (Yang et al. 2019), RoBERTa (Liu et al. 2019) have achieved significant improvements on many tasks. This paper utilizes XLNet (Yang et al. 2019) as the backbone and propose our approach to study the commonsense question answering problem.

Graph Neural Networks for NLP Recently, Graph Neural Networks (GNN) has been utilized widely in NLP. For example, Sun et al. (2019) utilizes Graph Convolutional Networks (GCN) to jointly extract entity and relation. Zhang, Qi, and Manning (2018) applies GNN to relation extraction over pruned dependency trees and achieves remarkable improvements. GNN has also been applied into multi-hop reading comprehension tasks (Tu et al. 2019, Kundu et al. 2019, Jiang et al. 2019). This paper utilizes GCN to represent graph nodes by utilizing the graph structure information, followed by graph attention which aggregates the graph representations to make the prediction.

Conclusion and Future Work

In this work, we focus on commonsense question answering and select CommonsenseQA (Talmor et al. 2019) dataset as the testbed. We propose an approach consisting of knowledge extraction and graph-based reasoning. In the knowledge extraction part, we extract evidence from heterogeneous external knowledge including structured knowledge source ConceptNet and Wikipedia plain texts. We construct graphs for both sources to utilize the relational structures in both sources. In the graph-based reasoning part, we propose a graph-based approach consisting of graph-based contextual word representation learning module and graph-based inference module. The first module utilizes graph structural information to re-define the distance between words for learning better contextual word representations. The second module adopts Graph Convolutional Net-work to encode neighbor information into the representations of nodes, followed by a graph attention mechanism for evidence aggregation to infer final answers. Experiments show our model can achieve significant improvement and achieve a new state-of-the-art on the CommonsenseQA leaderboard.

In future work, we will add more heterogeneous external knowledge sources and improve the reasoning module in our approach to further achieve improvements.
[Ye et al. 2019] Ye, Z.-X.; Chen, Q.; Wang, W.; and Ling, Z.-H. 2019. Align, mask and select: A simple method for incorporating commonsense knowledge into language representation models. CoRR abs/1908.06725.

[Zellers et al. 2018] Zellers, R.; Bisk, Y.; Schwartz, R.; and Choi, Y. 2018. SWAG: A large-scale adversarial dataset for grounded commonsense inference. In Proc. of EMNLP, 93–104.

[Zellers et al. 2019] Zellers, R.; Holtzman, A.; Bisk, Y.; Farhadi, A.; and Choi, Y. 2019. Hellaswag: Can a machine really finish your sentence? In Proc. of ACL, 4791–4800.

[Zhang, Qi, and Manning 2018] Zhang, Y.; Qi, P.; and Manning, C. D. 2018. Graph convolution over pruned dependency trees improves relation extraction. In Proc. of EMNLP, 2205–2215.