Knowledge-Based Commonsense Validation via Graph Convolutional Networks

GENG LIN 16307100060@fudan.edu.cn,
RUIPU LUO 16307130247@fudan.edu.cn,
CHENG ZHONG 16307110259@fudan.edu.cn,
School of Data Science,
Fudan University

Abstract

Introducing common sense to natural language understanding systems has received increasing research attention. The introduction of a common sense recognition system in a natural language processing system not only can help the system to distinguish unreasonable recognition results, also can increasing the confidence of the sentences generated by the machine. We add external knowledge such as common sense interpretation and knowledge maps, so that the model can better understand common sense. And our method performs better on a given data set than other traditional methods and BERT fine-tune model.

1 Introduction

Commonsense Knowledge (CS Know.) And Commonsense Reasoning are two important bottlenecks in machine intelligence. In 2015, Davis and Marcus (Davis and Marcus) pointed out the challenges of commonsense reasoning: Spans from difficulties in understanding and formulating commonsense knowledge for specific or general domains to complexities in various forms of reasoning and their integration for problem solving. Commonsense Reasoning is one of the most important and challenging topics in commonsense research. Existing tasks include traditional multi-choice tasks, generation tasks including explanation generation and commonsense generation and so on (Storks et al., 2019). With the development of commonsense research, many variants of tasks have emerged.

In this task, we using the dataset created by Cunxiang Wang and his collaborators (Wang et al., 2019).Based on the dataset given by SemEval2020, our research focus on commonsense explanation selection and commonsense explanation generation, two sub-tasks of Commonsense Reasoning. In a commonsense explanation selection task, a statement against commonsense is given, and one is asked to select an explanation from 3 choices that is most likely to be the explanation1. And generation task require a proper generated explanation for the statement against commonsense. An example of a commonsense explanation selection is shown in Figure 1.

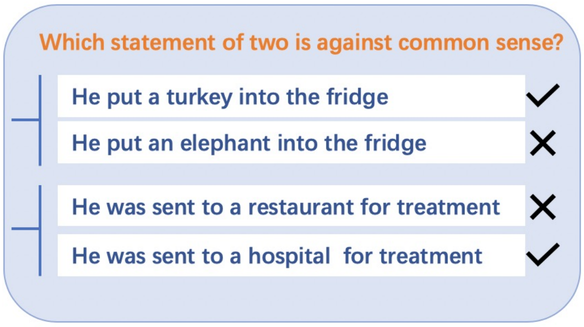

Figure 1: An example of a commonsense explanation selection(Wang et al., 2019)

Based on previous work and studies, for explanation selection task, we propose a method that combine powerful language model with graph information from knowledge bases.

In summary, The contribution of this paper is that we have added external knowledge to assist in the classification of common sense sentences. The information in the knowledge graph is obtained by extracting the entities in the sentence and then performing sub-graph matching in the knowledge graph. And we add explanations about common sense to assist classification.

2 Related Work

In researches of commonsense reasoning, varies of methods have proved their power. In recent researches, the pre-trained contextual models including BERT (Devlin et al.), XLNet (Yang et al.), RoBERTa (Devlin et al.) have greatly increased the performance of models in Commonsense tasks.

The current models mainly combine these new types of language models and use the huge corpus information in them to better combining linguistic information and improve model performance. These language models perform well in generation tasks too. Our work was carried out on Bert. In addition, the use of external knowledge outside has gradually become mainstream in prior studies. Many researches are combined with corpora to fine-tune the model with external knowledge. For example, Wikipedia contains a lot of commonsense information and relational data, which are widely used in Commonsense researches.

Knowledge bases also play an important role. The structural feature of knowledge base has special characteristics. There are many well-known algorithms in the field of graph data research to calculating the semantic information contained in the graph (Youngmann et al., 2019). At present, with the development of Commonsense research, some knowledge bases are widely used in Commonsense research, such as ConceptNet (Liu and Singh, 2004), DBpedia (Auer et al., 2007) and YAGO (Kasneci et al., 2006). ConceptNet is a famous semantic network, which is originated from the crowd sourcing project Open Mind Common Sense (Liu and Singh, 2002), which was launched in 1999 at the MIT Media Lab. Such knowledge bases have make great contribution to commonsense reasoning. Even some benchmarks are created from knowledge bases (Talmor et al., 2018).In the study of KagNet (Lin et al., 2019), the author utilized GCN network (Kipf and Welling, 2016) to combine the information on the knowledge base, then encoded the nodes, and finally obtained the graph representation. This research topped in the Commonsense QA competition organized by the Tel Aviv University. In another novel work (Lv et al., 2019), the author constructed a graph of the extracted evidence based on the information in ConceptNet, and then combined with GCN, and finally obtained the graph representation. It can be found that the knowledge base contains important commonsense information. Our research also extracts information from ConceptNet.

3 Methodology

3.1 Problem Defination

Each piece of data in the dataset is annotated with a label $\boldsymbol{l}$ and two sentences $\boldsymbol{S_1}, \boldsymbol{S_2}$ with common sense expressions where $\boldsymbol{S_1} = \{\boldsymbol{w_{11}}, \boldsymbol{w_{12}}, \boldsymbol{w_{13}} \ldots\}$and $\boldsymbol{S_2} = \{\boldsymbol{w_{21}}, \boldsymbol{w_{22}}, \boldsymbol{w_{23}}, \ldots\}$. Two sentences are very similar which in the same syntactic structure and differ by only few words, but only one of them makes sense while the other does not.Our goal is to choose which sentence is more make sense.

In order to better judge which sentence is more in line with common sense, we have introduced an external explanation of this common sense, which can be expressed as $\boldsymbol{E}$. In addition, we introduce external knowledge of the knowledge graph(In this task, we use ConceptNet denoted as $\boldsymbol{KG}$). In order to use this knowledge, we need entities in the sentence. We represent these entities as $\boldsymbol{e_1}, \boldsymbol{e_2}, \boldsymbol{e_3}, \ldots$.

If we named our model as $\boldsymbol{MODEL}$, the probability of S_1 be predicted as true sentence is

$$\boldsymbol{P}(S_1 \ is \ true) = \boldsymbol{MODEL}(S_1, S_2, E, KG)$$

So, the problem can be represented as

$$\arg\max(\boldsymbol{P}(S_1 \ is \ true), 1 - \boldsymbol{P}(S_1 \ is \ true))$$

3.2 Model Architecture

The whole model architecture is shown in Figure 2. The $\boldsymbol{input}$ is two sentence $\boldsymbol{S_1}, \boldsymbol{S_2}$ and the explanation(3.3) of common sense $\boldsymbol{E}$. Each circle point represents a word in a sentence. Blue points means entities in sentences, and black points represent words in explanation in sentence.

Let the text which splicing three parts $\boldsymbol{S_1}, \boldsymbol{S_2}, \boldsymbol{E}$ be BERT model's input. In the subsequent training, we will fix the parameters of the first 11 layers of BERT pre-trained model, and fine-tune the parameters of the last layer of BERT pre-trained model. The output of fine-tuned BERT model is a 768 dimension vector which represents the meaning of sentence 1, sentence 2 and explanation. We denoted the vector as $\boldsymbol{b}$.

In order to use the knowledge graph of common sense, ConceptNet, we first extract the entities in two sentence, and locate them in ConceptNet. Then we extract the sub graph which contains these entities and their $\boldsymbol{N}$ neighbors. Let the sub graph and the entities word embedding be $\boldsymbol{l}$ layers $\boldsymbol{GCN}$'s input, and the output of $\boldsymbol{GCN}$ can be denoted as $\boldsymbol{g}$

Let $\boldsymbol{h}$ denote the concatenation of $\boldsymbol{b}$ and $\boldsymbol{g}$,

$$\boldsymbol{h} = [\boldsymbol{b}, \boldsymbol{g}]$$

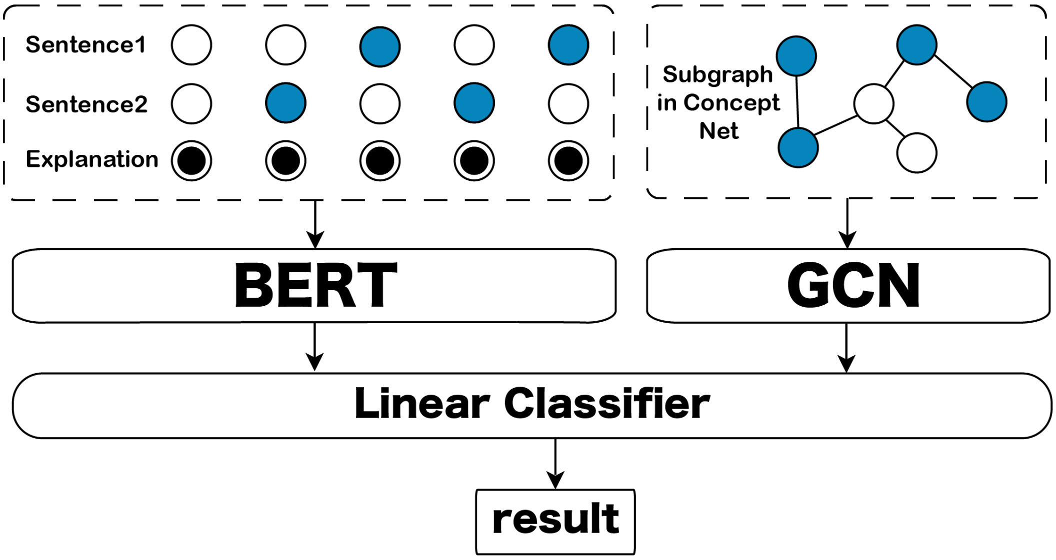

Figure 2: Whole model architecture. A blue point represent an entity in sentence and KG.

therefore vector $\boldsymbol{h}$ has contained all information from original sentences, explanation, and ConceptNet. We take the vector as the input of the linear classification layer, and finally get the classification result.

3.3 Explanation Information

In (Wang et al., 2019), two tasks about common sense understanding was proposed. Task 2 is to find an explanation for the establishment of common sense. After comparison, it is found that the data sets used in task 1 and task 2 are the same, so we can use the interpretation of common sense in each data set in task 2 as external knowledge. Added to the judgment of auxiliary task 1. An example can be seen in Figure 3.

Example of Explanation	
S_1	The light was too dim so I began reading.
S_2	The light was too dim so I stopped reading
E	Dim light is not suitable for reading

Figure 3: An example of a pair sentences with an explanation

3.4 Entities and External Knowledge

3.4.1 Extract Entities

Through the observation of the data set, we find that the text belongs to the open domain, not a specific domain, so in order to extract the entities, we choose to use the open domain text extraction method. Here we have chosen the best open source extraction tool OPENIE5[1].This tool will extract the entities in the sentence and the relationships between the entities as a triple,such as $\{Entity_1, Entity_2, relation\}$. We only need to use the entities extracted from it. An example of extraction can be seen in Figure 4. But sometimes

Entity Extraction	
Sentence	He dipped his chips in salsa.
Entity1	chip
Entity2	salsa

Figure 4: An example of entity extraction using OPENIE5

the extraction tool cannot extract entities, so we also extract all the nouns that appear in the entire sentence as entities.

3.4.2 External Knowledge: ConceptNet

After we get the $entities$, we find them in ConceptNet and find the N neighbor nodes of these $entities$. These entity nodes and neighbor nodes form a sub graph, and We extract this sub graph and

[1]https://github.com/dair-iitd/OpenIE-standalone

represent it as a symmetrical adjacency matrix $\boldsymbol{A}$. The input of our $\boldsymbol{GCN}$ layer is adjacency matrix $\boldsymbol{A}$ and the embedding matrix of these entities and neighbor nodes in ConceptNet denoted as $\boldsymbol{X_{embed}}$. Therefore, our $\boldsymbol{GCN}$ layer can be represented as follow,

$$\boldsymbol{g = \widetilde{A}ReLU(\widetilde{A}X_{embed}W^{(0)})W^{(1)}}$$

Here, $\widetilde{\boldsymbol{A}}$ is the normalized adjacency matrix of the sub graph from ConceptNet. $\boldsymbol{W^{(0)}}$ is an input-to-hidden weight matrix for a hidden layer. $\boldsymbol{W^{(0)}}$ is an hidden-to-output weight matrix. The output vector $\boldsymbol{g}$ contain the information from ConceptNet.

4 Experiments

We introduce our commonsense data set (from the course), our experimental setup, and the experimental results and analysis of the results.

4.1 Dataset

The entire common sense data set contains a total of 10,785 pieces of data. Each piece of data contains two sentences describing the same common sense and one label. One sentence is consistent with common sense, and the label indicates which sentence is correct.The explanation of common sense is derived from a data set of task2 in (Wang et al., 2019). After observation, we found that it can correspond to our task, so there are 10,785 explanations about common sense.In order to compare the performance of different methods and find the best model during the training, we divided the data set into a training set, development set and test set (7007/ 1761/ 2017). We will compare different methods on the test set.

For external knowledge, we use the open source ConceptNet (Liu and Singh, 2004). ConceptNet is a semantic network built by MIT, which contains a lot of information that computers should know about this world, which helps computers do better searches, answer questions, and understand human intentions. It consists of nodes that represent concepts. These concepts are expressed in the form of words or phrases in natural language, and the relationships between these concepts are marked.

4.2 Experiment Setup

We implemented the Our model using Pytorch framework. We used 4 2080-Ti GPUs for parallel training. We choose the "bert-base-uncased" model for fine-tuning. The adjacency matrix we constructed is 500 dimensional, which means that we searched the knowledge graph for 500 nodes containing the entities in the sentence as entity subgraphs. When the $\boldsymbol{GCN}$ encodes an entity subgraph, we select 100-dimensional word embedding, and the output dimension is 100. We optimized the model with Adam, the initial learning rate is set to 1e-5, and the Adam epsilon is 1e-8. We trained the model for 5 epochs, and per GPU's train batch size is 4. Random number seed we set to 42. The total training time is 40 minutes.

4.3 Result

We used the script provided in the 'Final PJ' folder to evaluate the accuracy of different baseline methods as well as our Graph Knowledge-Based with Commonsense Explanation methods.

We use $\boldsymbol{SVM}, \boldsymbol{LSTM}, \boldsymbol{Bert-finetune}$ as our baseline. The development and test set accuracy of all methods is shown in the Table 1.

Experiment Result			
Method		**Dev Acc**	**Test Acc**
Baseline	**SVM**	0.5423	0.5274
	LSTM	0.7357	0.6984
	BERT	0.8489	0.8076
Ours	**BERT +Explain**	0.8845	0.8582
	BERT +KB	0.8604	0.8193
	BERT +Explain +KB	**0.8914**	**0.8726**

Table 1: baseline and our model's experimental results

Among all methods, our model, BERT + Explain+ KB , achieved the best results, and the accuracy on the test set reached **0.8726**. And we can find that among the three baselines, the best performing is the BERT fine-tuning, which has an accuracy rate of **0.8076** on the test set. Therefore, the performance of our model far exceeds the best performing baseline model's performance.

In order to compare the improvement effect of two different external knowledge on the whole task, we also did experiments for adding only the knowledge graph and adding only commonsense explanation. Through the experimental results, it is found that adding commonsense explanation to adding the knowledge graph has a greater impact on the accuracy of the entire task.

5 Conclusion and Future Work

We proposed a knowledge based commonsense validation model using graph conlolutional network. In addition to adding an open source knowledge graph, we also added an explanation about common sense. From the experimental results, this makes the model better understand the meaning of this common sense.

For future research, we will further investigate how to find corresponding commonsense from existing external knowledge. We believe that increasing the utilization of external knowledge is the key to improving the accuracy of classification. The use of ConceptNet in this article is not very efficient. So we hope to explore some more direct ways of using external knowledge, which may be more helpful for improving performance.

References

Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. 2007. Dbpedia: A nucleus for a web of open data. In *The semantic web*, pages 722–735. Springer.

Ernest Davis and Gary Marcus. Commonsense reasoning and commonsense knowledge in artificial intelligence. *Communications of the Acm*, 58(9):92–103.

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

Gjergji Kasneci, Fabian Suchanek, and Gerhard Weikum. 2006. Yago-a core of semantic knowledge.

Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

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. 2002. Makebelieve: Using commonsense knowledge to generate stories. In *AAAI/IAAI*, pages 957–958.

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

Shangwen Lv, Daya Guo, Jingjing Xu, Duyu Tang, Nan Duan, Ming Gong, Linjun Shou, Daxin Jiang, Guihong Cao, and Songlin Hu. 2019. Graph-based reasoning over heterogeneous external knowledge for commonsense question answering. *arXiv preprint arXiv:1909.05311*.

Shane Storks, Qiaozi Gao, and Joyce Y Chai. 2019. Commonsense reasoning for natural language understanding: A survey of benchmarks, resources, and approaches. *arXiv preprint arXiv:1904.01172*.

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

Cunxiang Wang, Shuailong Liang, Yue Zhang, Xiaonan Li, and Tian Gao. 2019. Does it make sense? and why? a pilot study for sense making and explanation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. Xlnet: Generalized autoregressive pretraining for language understanding.

Brit Youngmann, Tova Milo, and Amit Somech. 2019. Boosting simrank with semantics. In *EDBT*, pages 37–48.