PGL at TextGraphs 2020 Shared Task: Explanation Regeneration using Language and Graph Learning Methods

Weibin Li, Yuxiang Lu, Zhengjie Huang, Jiaxiang Liu
Weiyue Su, Shikun Feng, Yu Sun
Baidu Inc., China
{liweibin02,luyuxiang,huangzhengjie,liujiaxiang}@baidu.com
{suweiyue,fengshikun01,sunyu02}@baidu.com

Abstract

This paper describes the system designed by the Baidu PGL Team which achieved the first place in the TextGraphs 2020 Shared Task. The task focuses on generating explanations for elementary science questions. Given a question and its corresponding correct answer, we are asked to select the facts that can explain why the answer is correct for that question and answering (QA) from a large knowledge base. To address this problem, we use a pre-trained language model to recall the top-K relevant explanations for each question. Then, we adopt a re-ranking approach based on a pre-trained language model to rank the candidate explanations. To further improve the rankings, we also develop an architecture consisting both powerful pre-trained transformers and GNNs to tackle the multi-hop inference problem. The official evaluation shows that, our system can outperform the second best system by 1.91 points.

1 Introduction

The TextGraphs 2020 Shared Task on Explanation Regeneration (Jansen and Ustalov, 2020) asks participants to develop methods to reconstruct gold explanations for elementary science questions. Concretely, given an elementary science question and its corresponding correct answer, the system need to perform the multi-hop inference and rank a set of explanatory facts that are expected to explain why the answer is correct from a large knowledge base.

Multi-hop inference is the task of combining more than one piece of information to solve a reasoning task, such as question answering. Multi-hop inference or information aggregation has been shown to be extremely challenging (Jansen, 2018), especially for the case here, where current estimates suggest that an average of 4 to 6 sentences are required to answer and explain a given question. An example is shown in Figure 1.

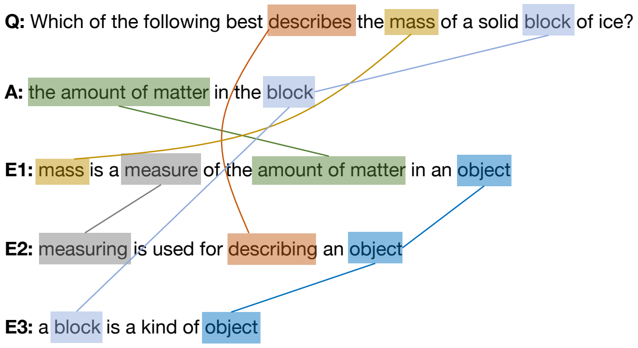
Q: Which of the following best describes the mass of a solid block of ice?
A: the amount of matter in the block
E1: mass is a measure of the amount of matter in an object
E2: measuring is used for describing an object
E3: a block is a kind of object

Figure 1: A subgraph of explanation sentences that explains why the answer is correct for the question.

In the TextGraphs 2020 Shared Task, we not only need to consider extracting the semantic information between the question and each explanation, but also need to take the structural relationship between the

Proceedings of the Graph-based Methods for Natural Language Processing (TextGraphs), pages 98–102
Barcelona, Spain (Online), December 13, 2020

explanations into account. Therefore, we adopt a pipeline architecture to address the problem. First, we use a pre-trained language model to recall the top-K relevant explanations for each question. Then, we adopt a re-ranking approach based on another pre-trained language model to rank the candidate explanations. Finally, to further improve the rankings, we also develop an architecture utilizing the power of pre-trained transformers (Vaswani et al., 2017) and graph neural networks (GNNs) (Kipf and Welling, 2016) to tackle the multi-hop inference problem. We also adopt a Virtual Adversarial Training (Takeru et al., 2018) method to train our model and got a slight improvement.

The rest of the paper is organized as follows. In Section 2, we will briefly describe the task, the dataset and the evaluation metrics of the task. Section 3 shows the details of our approach. Our experiments will be shown in Section 4.

2 Task

Task. As described in Section 1, the TextGraphs 2020 Shared Task focuses on selecting a set of explanation sentences that can explain the answer of a question, which can be regarded as a ranking task. Concretely, given an elementary science question, its corresponding correct answer and a set of explanation sentences, the goal is to determine whether an explanation sentence is the reason for the QA.

Corpus. The data used in this shared task comes from the WorldTree V2 corpus (Xie et al., 2020). The dataset includes approximately 4400 standardized elementary and middle school science exam questions (3rd through 9th grade). Each example in the WorldTree V2 corpus contains detailed annotation stating whether a fact is a part of the explanation for that question. For each explanation, the WorldTree V2 corpus also includes annotation for how important each fact is towards the explanation.

Evaluation. Explanation reconstruction performance is evaluated in terms of mean average precision (MAP) by comparing the ranked list of facts with the gold explanation. Therefore, it is intuitive for us to regard the task as a ranking problem.

3 Approach

Our system consists of two major components. The first part is an information retrieval (IR) system based on the pre-trained language model to retrieve the top-K relevant explanation sentences from the whole knowledge base. The second part consists of two modules, including a pointwise ranking module to rank candidate facts and a graph-based module to counter the problem of multi-hop inference.

3.1 Retrieval

Recently, pre-trained language models (Devlin et al., 2018; Liu et al., 2019; Lan et al., 2019; Sun et al., 2020) have achieved state-of-the-art results in various language understanding tasks such as question answering (Rajpurkar et al., 2016; Khashabi et al., 2018). For our IR system, we use ERNIE 2.0 (Sun et al., 2020), the world's first model to score over 90 in terms of the macro-average score on GLUE benchmark (Wang et al., 2018), as our retriever. We concatenate the question, the correct answer and the explanation sentence as input of the retriever which will return a score to determine whether an explanation sentence is relevant to that question. Then for each question, we can get the top-K ranked facts from the corpus. Although a simple tf-idf based retriever can obtain the top-K ranked facts in a shorter time, its result is not very effective compared with the pre-trained model, as shown in Table 1. Since the pre-trained model already has strong semantic representation capabilities, it can achieve an excellent result on 5000 steps fine-tuning within two hours. Details can be found in Section 4.1.

Retriever	MAP@top100	Oracle MAP@top100
TF-IDF	25.49%	50.78%
Ours	48.80%	92.03%

Table 1: The recall result of different retrievers on the development set. The **Oracle MAP@top100** is the upper bound MAP score where all the relevant facts in these 100 candidate facts are ranked first.

3.2 Ranking

Our re-ranking component consists of two modules. Since we only fine-tune the retriever for 5000 steps, it can be still improved by the pre-trained model. Therefore, we use another pre-trained model based on ERNIE 2.0 (Sun et al., 2020) to re-rank the candidate explanation sentences from the retrieval stage. We significantly improve the performance on the task, outperforming the retriever by more than 10% of MAP. However, we found that there are many facts that have lexical overlap with the question, but they are not the reason for the QA. On the contrary, some key facts that have no lexical overlap with the question are ranked low. This kind of key facts are usually the explanation of other relevant facts, rather than the direct explanation of the question. Since each sample is only composed of a question, a correct answer and an explanation sentence, it is difficult for the retriever to learn the correlation between the candidate facts.

To address this problem, we utilize the graph neural networks (GNNs) to learn the correlation between the candidate facts. Graph neural networks (GNNs) are recursive neural networks for modeling the graph structure. Concretely, the graph structure here is the correlation between the candidate facts. As shown in Figure 1, **E1** explains the word ***mass*** for the question directly. **E2** explains the word ***measure*** for the **E1**. Therefore, **E2** can be regarded as a second order neighbor of the question, and we want to learn such relation with help of GNNs. Modern GNNs follow a neighborhood aggregation strategy, where we iteratively update the representation of a node by aggregating representations of its neighbors. In an attempt to integrate the powerful language understanding ability into graph learning, we present a graph aggregator with pre-trained transformers. Figure 2 shows the details of the architecture.

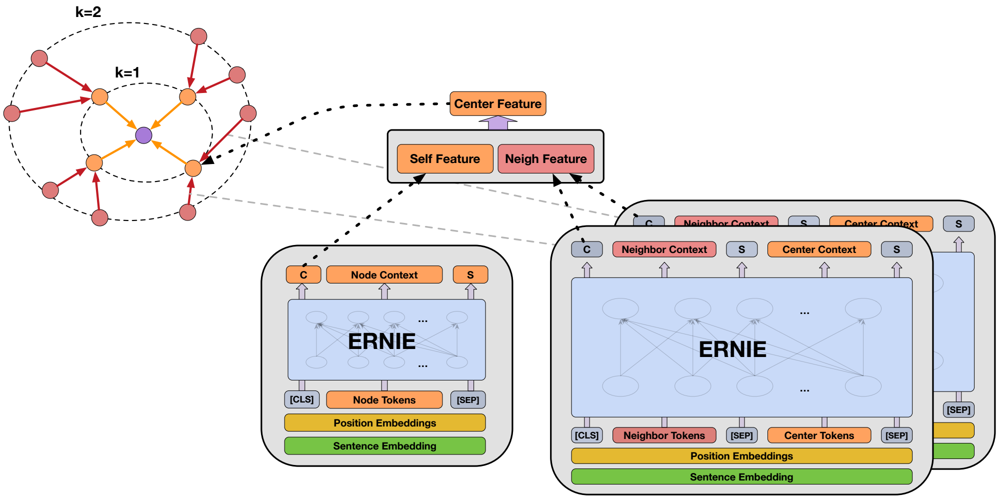

Figure 2: The overview of the architecture that integrates the powerful language understanding ability into graph learning.

As GraphSAGE-like (Hamilton et al., 2017) aggregation function only aggregate neighbors using simple operators, such as sum, mean or max. There's no direct interaction between the center node and each neighbor. In a text graph, node (sentence) interaction should not be limited in node-level (sentence-level) embedding. It should take the token-level (word-level) interaction between two nodes into account. In an attempt to make token-level interaction possible, we apply ERNIE on the edges of the graph by concatenating raw text tokens of the node pairs on the sampled edges (PGL, 2020). As shown in Figure 2, instead of obtaining the neigh feature directly from the neighbor sentence, we get the neighbor features by the interaction between the center tokens and the neighbor tokens. Then, the **[CLS]** embedding will be taken as the neigh feature.

To train the model we described above, we need to construct the edges between explanation facts. The K candidate explanation sentences for a question and a answer are regarded as nodes to form a graph. Here, edges can be a result of lexical overlap between explanation sentences. But we find that using this method will result in a very dense graph, and hence, each node in the graph is linked with many neighbors and most of them are irrelevant to the QA.

To alleviate this problem, we adopt a pairwise binary classification system, to score the explanation fact pairs in the candidate set for each question. We use the pre-trained language model to judge whether the explanation fact pair is relevant to the question and the answer. If both the two explanation sentences are relevant to the question and the answer, the label is 1, otherwise the label is 0. Then we rank all the explanation fact pairs by the score and select the top-M pairs as the edges of the text graph for each question. We then feed them to the text graph model we described above, and regard it as a node classification task.

4 Experiments

4.1 Model Configuration

In this work, the pre-trained language model we used to encode the text is ERNIE 2.0 (Sun et al., 2020), which is considered to be an expressive powerful model. For all experiments, the learning rate of the ERNIE encoder was initialized to $1e^{-4}$, batch size is 64 and the maximum sequence length is 128. We used the Adam optimizer with linear learning rate decay. In the retrieval phase, we fine-tuned the model for 5000 steps on a NVIDIA Tesla V100 (32GB GPU) machine. In the ranking phase, the pre-trained model was fine-tuned for 1 epoch with virtual adversarial training. To generate the edges for applying GNNs with the pre-trained language model to tackle multi-hop inference problem, we fine-tuned the ERNIE model for 5000 steps and selected the top 20 explanation sentence pairs as the edges for each question.

For evaluation, we select the top 100 ranked facts from the retrieval phase, and we found that the oracle MAP score can reach 92.03% with top 100 ranked facts, as shown in Table 1. We concatenate the correct answer choice with the question, because we found that adding the wrong options can mislead the model and leads to a lower MAP score. It is intuitive since the wrong options are not necessary to answer the question.

4.2 Experiment Results

We report the tf-idf based ranking scores as the baseline. From the Table 2 we can see that, though the tf-idf method can quickly score all the facts, its MAP score is very low compared with the ERNIE retriever. The ERNIE Re-ranker can significantly improve the performance on the task, which outperforms the retriever for more than 10% points of MAP score. The ERNIE graphsage model can also improve the performance of the Re-ranker. To further improve the performance on the leaderboard, we run our ERNIE Re-ranker model for three times and then ensemble them to get a better performance.

The performance of hidden test set of our final model is shown in Table 3. Our submission achieved the first place in TextGraphs 2020 Shared Task.

Model	**MAP@top50**	**MAP@top100**
TF-IDF	25.16%	25.49%
ERNIE Retriever	48.17%	48.80%
ERNIE Re-ranker	58.73%	59.16%
+ ERNIE Graphsage	59.53%	59.99%
ensemble ranking	61.58%	61.98%

Table 2: MAP score on the development set.

Model/participant	**MAP**
Our model	60.33%
alvysinger	58.43%
aisys	52.33%

Table 3: MAP score on the test set.

5 Conclusion

We proposed our approach to the shared task on “Multi-hop Inference Explanation Regeneration”. Our system consists of a pre-trained model-based retriever and a graph-based pre-trained model for the re-ranking phase, and achieved the first place in TextGraphs 2020 Shared Task.

References

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*.

Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. In *Advances in neural information processing systems*, pages 1024–1034.

Peter Jansen and Dmitry Ustalov. 2020. TextGraphs 2020 Shared Task on Multi-Hop Inference for Explanation Regeneration. In *Proceedings of the Graph-based Methods for Natural Language Processing (TextGraphs)*. Association for Computational Linguistics.

Peter Jansen. 2018. Multi-hop inference for sentence-level textgraphs: How challenging is meaningfully combining information for science question answering? *arXiv preprint arXiv:1805.11267*.

Daniel Khashabi, Snigdha Chaturvedi, Michael Roth, Shyam Upadhyay, and Dan Roth. 2018. Looking beyond the surface: A challenge set for reading comprehension over multiple sentences. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 252–262.

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

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. In *International Conference on Learning Representations*.

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*.

Team PGL. 2020. Erniesage: Ernie sample aggregate. `https://github.com/PaddlePaddle/PGL/tree/master/examples/erniesage`.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392.

Yu Sun, Shuohuan Wang, Yu-Kun Li, Shikun Feng, Hao Tian, Hua Wu, and Haifeng Wang. 2020. Ernie 2.0: A continual pre-training framework for language understanding. In *AAAI*, pages 8968–8975.

Miyato Takeru, Maeda Shin-Ichi, Ishii Shin, and Koyama Masanori. 2018. Virtual adversarial training: A regularization method for supervised and semi-supervised learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355.

Zhengnan Xie, Sebastian Thiem, Jaycie Martin, Elizabeth Wainwright, Steven Marmorstein, and Peter Jansen. 2020. WorldTree v2: A corpus of science-domain structured explanations and inference patterns supporting multi-hop inference. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 5456–5473, Marseille, France, May. European Language Resources Association.