DeepBlueAI at TextGraphs 2021 Shared Task: Treating Multi-Hop Inference Explanation Regeneration as A Ranking Problem

Chunguang Pan **Bingyan Song** **Zhipeng Luo**
DeepBlue Technology (Shanghai) Co., Ltd
{panchg, songby, luozp}@deepblueai.com

Abstract

This paper describes the winning system for TextGraphs 2021 shared task: Multi-hop inference explanation regeneration. Given a question and its corresponding correct answer, this task aims 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 and accelerate training as well, our strategy includes two steps. First, fine-tuning pre-trained language models (PLMs) with triplet loss to recall top-K relevant facts for each question and answer pair. Then, adopting the same architecture to train the re-ranking model to rank the top-K candidates. To further improve the performance, we average the results from models based on different PLMs (e.g., RoBERTa) and different parameter settings to make the final predictions. The official evaluation shows that, our system can outperform the second best system by 4.93 points, which proves the effectiveness of our system. Our code has been open source, address is https://github.com/DeepBlueAI/TextGraphs-15

1 Introduction

Multi-hop inference is the task of doing inference by combining more than one piece of information, such as question answering (Jansen and Ustalov, 2019). The TextGraphs 2021 Shared Task on **Multi-Hop Inference Explanation Regeneration** focuses on the theme of determining relevance versus completeness in large multi-hop explanations, which asks participants to rank how likely table row sentences are to be a part of a given explanation. 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. An example is shown in Figure 1.

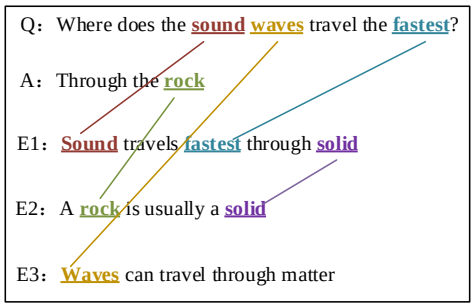

Figure 1: A multi-hop inference example which can explain why the answer is correct for the question.

A number of contemporary challenges exist in performing multi-hop inference for question answering (Thayaparan et al., 2020), including semantic drift, long inference chains, etc. Several Multi-hop inference shared tasks have been conducted in the past few years (Jansen and Ustalov, 2019, 2020), and methods based on large pre-trained language models (PLMs) such as BERT (Das et al., 2019; Chia et al., 2019), RoBERTa (Pawate et al., 2020) and ERNIE (Li et al., 2020) are proposed.

In this paper, we describe the system that we submitted to the TextGraphs 2021 shared task on Multi-Hop Inference Explanation Regeneration. There are two main parts of our system. First, we use a pre-trained language model-based method to recall the top-K relevant explanations for each question. Second, we adopt the same model architecture to re-rank the top-K candidates to do the final prediction.

When determine whether an explanation sentence is relevant to the question, the previous works (Das et al., 2019; Li et al., 2020) constructed a pair of explanations with the QA (questions with corresponding answers) sentence as the input of the PLMs. To reduce the amount of calculation and accelerate training, instead of using all the explanations from the given table, we propose to fine-tune PLMs with triplet loss (Schroff et al., 2015), a loss

Proceedings of the Fifteenth Workshop on Graph-Based Methods for Natural Language Processing (TextGraphs-15), pages 166–170

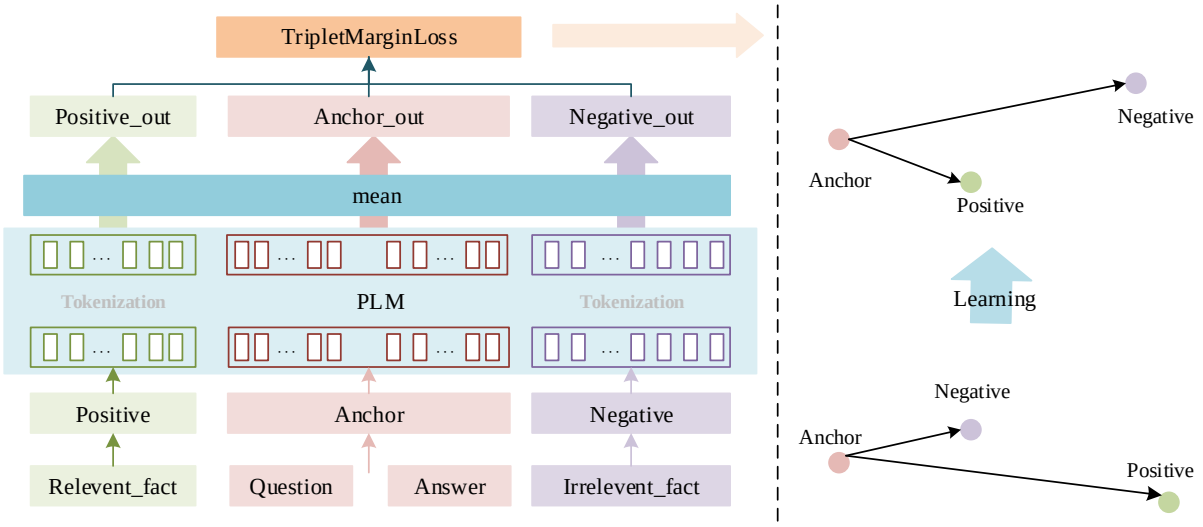

Figure 2: The architecture of the proposed model.

function where a baseline (anchor) input is compared to a positive (true) input and a negative (false) input. For choosing samples as the negative input, we design several ways which will be introduced in Section 3. Experiments on the given dataset show the effectiveness of our model and we rank first in this task.

2 Background

Task Definition The explanation regeneration task supplies models with questions, their correct answers, the gold explanation authored by human annotators, as well as a knowledge base of explanations. From this, for a given question and its correct answer, the model must select a set of explanations from the knowledge base that explain why the answer is correct.

Dataset The data used in this shared task contains approximately 5,100 science exam questions drawn from the AI2 Reasoning Challenge (ARC) dataset (Clark et al., 2018), together with multi-fact explanations for their correct answers extracted from the WorldTree V2.1 explanation corpus (Xie et al., 2020; Jansen et al., 2018). Different from shared task in 2020 (Jansen and Ustalov, 2020), this year's dataset has been augmented with a new set of approximately 250k pre-release expert-generated relevancy ratings. The knowledge base supporting these questions and their explanations contains approximately 9,000 facts. These facts are a combination of scientific knowledge as well as commonsense/world knowledge.

Evaluation As mentioned in the official evaluation procedure of TextGraphs 2021, the participating systems are evaluated using Normalized Discounted Cumulative Gain (NDCG), a common measure of ranking quality. Therefore, it inspires us to think of this task as a ranking task.

3 Model Architectures

Our system consists of two major components. The first part is the retrieval procedure, which utilize the PLMs fine-tuned with triplet loss to recall top-K (K>100) relevant explanations. The second part is the re-ranking procedure, which use the same model architecture to rank the top-K candidates. The model architecture is shown in Figure 2.

3.1 Model

Inspired by the work of Schroff et al. (2015), we adopt the triplet loss in this task. The triplet loss minimizes the distance between an anchor and a positive, and maximizes the distance between the anchor and a negative. We treat the sentences of questions with corresponding answers as the anchor, the facts annotated with high reference value as positives. Both in retrieval procedure and re-ranking procedure, we generate three different negative samples for each positive and anchor pair, which will be discussed in Section 3.3.

After constructing triplet (an anchor, a positive, a negative), we put them into the PLMs (e.g., RoBERTa) to get their representations. These PLMs first tokenize the input sentences and then output the last layer embedding of each tokens. We average each token's embedding as the final representations for the positives, anchors and negatives,

which are denoted by e_p, e_a and e_n respectively. Then, the models can be fine-tuned by the triplet loss.

3.2 Triplet loss

After obtaining the embeddings of the triplet (an anchor (a), a positive (p) and a negative (n)), the triplet loss can be calculated as follow,

$$\mathcal{L}(a,p,n) = max\{d(e_a,e_p) - d(e_a,e_n) + \alpha, 0\} \quad (1)$$

$$d(x,y) = \|x-y\|_2 \quad (2)$$

α is a margin that is enforced between positive and negative pairs.

3.3 Training procedure

Retrieval First, we use the model introduced above to recall top-K relevant facts. In this step, for each anchor and positive pair, the negative samples are selected by three ways: 1) a sample which comes from the same table file with the positive one and does not annotated as the relevant one with the anchor; 2) a sample within the same mini-batch of positives and does not annotated as the relevant one with the anchor 3) a sample selected randomly among the facts irrelevant to the anchors.

Re-ranking After obtaining the top-K relevant facts, we train the re-ranking model with the same model architecture, yet use the another three different ways to select negative samples: 1) a sample within the top-K candidates but is irrelevant to the anchors; 2) a sample within top-100 candidates but irrelevant to the anchors; 3) a sample within the same mini-batch of positives but irrelevant to the anchors.

Ensembling Finally, to further improve the performance, we average different results from models based on different PLMs and random seeds.

4 Experiments

4.1 Parameter settings

All models are implemented based on the open source transformers library of hugging face (Wolf et al., 2020), which provides thousands of pre-trained models that can be quickly download and fine-tuned on specific tasks. The PLMs we used in this task are RoBERTa (Liu et al., 2019) and ERNIE 2.0 (Sun et al., 2020). For all the experiments, we set the batch size as 48 and set 15 epochs for both retrieval and re-ranking procedure. We use the Adam optimizer and create a schedule with a learning rate that decreases linearly from the initial lr set ($1e^{-5}$) in the optimizer to 0, after a warmup period during which it increases linearly from 0 to the initial lr set in the optimizer.

Method	NDCG
within the same mini-batch	0.7597
randomly	0.7621
within the same file	**0.7726**
all the three above	0.771

Table 1: The comparison between different ways of selecting negative samples

Methods	Recall
TF-IDF	0.7001
Ensemble Retriever	**0.97562**

Table 2: The comparison between different retrieval models

4.2 Ablation studies

Retrieval Since we have design three different ways to choose the negative samples during the retrieval procedure, we did experiments on the validation set to test whether these mechanisms valid or not. From Table 1, we find the most effective way is to choose the negative samples from the same table file with the positive one. Facts in the same table file have the same pattern.

Since for each question and answer pair, there are usually more than ten annotated relevant facts, we select the top-2000 ranked facts from the retrieval phrase, and we find that the NDCG score can reach 97.56% as shown in Table 2. Besides, though the TF-IDF method can quickly score all the facts, its NDCG score is very low compared with our retriever, which proves the effectiveness of our proposed method.

Re-ranking To re-rank the top-K candidates, we adopt the same model architecture. We compared the results of the proposed ensemble re-ranker with the TF-IDF baseline model and the proposed ensemble retriever on the test set, as shown in Table 3. We also set different top-K for calculating NDCG@K including 100, 500, 1000, and 2000. From Table 3, we can see that after re-ranking the top-K candidates, the model performance will be improved. Besides, as the increase of K value, the growth rate of NDCG gradually slows down.

Model	NDCG @100	NDCG@500	NDCG@1000	NDCG@2000
TF-IDF	0.5011	0.5271	0.5318	0.5352
Ensemble Retriever	0.7635	0.7819	0.7846	0.7857
Ensemble Re-ranker	0.8027	0.8171	0.8189	**0.8198**

Table 3: The final results compared with different models

4.3 Official Ranking

We submitted the scores predicted by the re-ranking model introduced above. The official ranking is presented in Table 4. We rank first in the task, 4.9% higher than the second place, which verifies the validity of our system.

Team	NDCG
DeepBlueAI	**0.8198**
RedDragonAI	0.7705
google-BERT	0.7003
huawei_noah	0.6831
tf-idf baseline	0.5010

Table 4: Leaderboard

5 Conclusion

In this paper, we propose a top performing approach for the task of multi-hop inference explanation regeneration. We fine-tune pre-trained language models with the triplet loss to accelerate training and design different ways for negative sampling. The same model architecture is utilized to recall the top-K candidates from all the facts and to re-rank the top-K relevant explanations for the final prediction. Experimental results show the effectiveness of the proposed method and we win the first place for the task.

References

Yew Ken Chia, Sam Witteveen, and Martin Andrews. 2019. Red dragon ai at textgraphs 2019 shared task: Language model assisted explanation generation. *arXiv preprint arXiv:1911.08976*.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*.

Rajarshi Das, Ameya Godbole, Manzil Zaheer, Shehzaad Dhuliawala, and Andrew McCallum. 2019. Chains-of-reasoning at textgraphs 2019 shared task: Reasoning over chains of facts for explainable multi-hop inference. In *Proceedings of the Thirteenth Workshop on Graph-Based Methods for Natural Language Processing (TextGraphs-13)*, pages 101–117.

Peter Jansen and Dmitry Ustalov. 2019. Textgraphs 2019 shared task on multi-hop inference for explanation regeneration. In *Proceedings of the Thirteenth Workshop on Graph-Based Methods for Natural Language Processing (TextGraphs-13)*, pages 63–77.

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)*, pages 85–97, Barcelona, Spain (Online). Association for Computational Linguistics.

Peter Jansen, Elizabeth Wainwright, Steven Marmorstein, and Clayton Morrison. 2018. WorldTree: A corpus of explanation graphs for elementary science questions supporting multi-hop inference. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Weibin Li, Yuxiang Lu, Zhengjie Huang, Weiyue Su, Jiaxiang Liu, Shikun Feng, and Yu Sun. 2020. Pgl at textgraphs 2020 shared task: Explanation regeneration using language and graph learning methods. In *Proceedings of the Graph-based Methods for Natural Language Processing (TextGraphs)*, pages 98–102.

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*.

Aditya Girish Pawate, Varun Madhavan, and Devansh Chandak. 2020. Chisquarex at textgraphs 2020 shared task: Leveraging pretrained language models for explanation regeneration. In *Proceedings of the Graph-based Methods for Natural Language Processing (TextGraphs)*, pages 103–108.

Florian Schroff, Dmitry Kalenichenko, and James Philbin. 2015. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823.

Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Hao Tian, Hua Wu, and Haifeng Wang. 2020. Ernie 2.0: A continual pre-training framework for language understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8968–8975.

Mokanarangan Thayaparan, Marco Valentino, and André Freitas. 2020. A survey on explainability in machine reading comprehension. *arXiv preprint arXiv:2010.00389*.

Thomas Wolf, Julien Chaumond, Lysandre Debut, Victor Sanh, Clement Delangue, Anthony Moi, Pierric Cistac, Morgan Funtowicz, Joe Davison, Sam Shleifer, et al. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45.

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. European Language Resources Association.