ChiSquareX at TextGraphs 2020 Shared Task: Leveraging Pre-trained Language Models for Explanation Regeneration*

Aditya Girish Pawate
IIT Kharagpur
adityagirish
pawate@gmail.com

Devansh Chandak
IIT Bombay
dchandak99@gmail.com

Varun Madhavan
IIT Kharagpur
varun.m.iitkgp
@gmail.com

Abstract

In this work, we describe the system developed by a group of undergraduates from the Indian Institutes of Technology, for the Shared Task at TextGraphs-14 on Multi-Hop Inference Explanation Regeneration (Jansen and Ustalov, 2020). The shared task required participants to develop methods to reconstruct gold explanations for elementary science questions from the WorldTree Corpus (Xie et al., 2020). Although our research was not funded by any organization and all the models were trained on freely available tools like Google Colab which restricted our computational capabilities, we have managed to achieve noteworthy results placing ourselves in the 4th place with a MAP score of 0.4902[1] in the evaluation leaderboard and 0.5062 MAP score on the post-evaluation-phase leaderboard using RoBERTa. We incorporated some of the methods proposed in the previous edition of Textgraphs-13 (Chia et al., 2019), which proved to be very effective, improved upon them, and built a model on top of it using powerful state-of-the-art pre-trained language models like RoBERTa (Liu et al., 2019), BART (Lewis et al., 2020), SciBERT (Beltagy et al., 2019) among others. Further optimization of our work can be done with the availability of better computational resources.

1 Introduction

The Shared Task is aimed at Multi-hop Inference for Explanation Regeneration. Participants are required to develop new and improve existing methods to reconstruct gold explanations for the WorldTree Corpus (Xie et al., 2020) of elementary science questions, their answers, and explanations.

Question: Which of the following is an example of an organism taking in nutrients? (A) a dog burying a bone (B) a girl eating an apple (C) an insect crawling on a leaf (D) a boy planting tomatoes
Answer: (B) a girl eating an apple
Gold Explanation Facts: 1) A girl means a human girl: Grounding 2) Humans are living organisms: Grounding 3) Eating is when an organism takes in nutrients in the form of food: Central 4) Fruits are kinds of foods: Grounding 5) An apple is a kind of fruit: Grounding
Irrelevant Explanation Facts: 1) Some flowers become fruits. 2) Fruit contains seeds. 3) living things live in their habitat. 4) Consumers eat other organisms

Table 1: An Example for Explanation Regeneration

The example highlights an instance for this task, where systems need to perform multi-hop inference to combine diverse information and identify relevant explanation sentences required to answer the specific question. The task provides a new and more challenging corpus of 9029 explanations and a set of gold explanations for each question and correct answer pair.

[1]Full, replicable code is available on Github for all methods described here, at https://github.com/dchandak99/TextGraphs-2020

This work is licensed under a Creative Commons Attribution 4.0 International Licence. Licence details: http://creativecommons.org/licenses/by/4.0/.

Proceedings of the Graph-based Methods for Natural Language Processing (TextGraphs), pages 103–108
Barcelona, Spain (Online), December 13, 2020

Team	MAP on leaderboard
Baidu PGL	0.6033
alvysinger	0.5843
aisys	0.5233
ChiSquareX	**0.4902**
Red Dragon	0.4793
mler	0.3367
dustalov (Baseline)	0.2344

Table 2: Final Leaderboard

Attributes	TG 2019	TG 2020
Questions	1680	4367
Explanations	4950	9029
Tables	62	81

Table 3: Dataset Comparison

2 Dataset

The dataset is the WorldTree Corpus V2.1(Xie et al., 2020) of Explanation Graphs and Inference Patterns supporting Multi-hop Inference (Februrary 2020 snapshot). It is a newer version of the dataset used in the TextGraphs-2019 (Jansen and Ustalov, 2019). The comparison between the two datasets is shown in Table 3.

3 Problem Review

The problem statement requires participants to build a system that, given a question and its answer choices, can identify the sentences that explain the answer given the question. This is a challenging task due to the presence of other irrelevant sentences in the corpora for the given question, which have equally significant lexical and semantic overlap as the correct ones (Fried et al., 2015). When a more classical graph theory approach using the semantic overlap of explanations and questions is tried, it leads to the problem of semantic drift (Jansen, 2018). More classic graph methods were attempted in (Kwon et al., 2018), where the challenge of semantic drift in multi-hop inference was analyzed, and the effectiveness of information extraction methods was demonstrated. Also, approaching the question as a language generation task is not effective and the current state-of-the-art models (Dušek et al., 2020) are not capable of generating the exact explanations as required by this task. So this task can easily be transformed into a sentence ranking problem in which we need to rank the relevant facts over all other given facts present in the corpus. The evaluation metric used for the task is the widely used and robust mean average precision (MAP) metric.

We have explained a few initial experiments that were undertaken in Section 4.1, followed by the pre-processing methods we incorporated in Section 4.2. We have then discussed our models in Sections 4.3 through 4.6. We have finally shown all our results and discussions in Section 5 followed by the conclusion and acknowledgments.

4 Model

4.1 Initial Experiments

We used the pure textual form of each explanation, problem and correct answer, rather than using a semi-structured form given in the column-oriented files provided in the dataset. Initially, we just reduced the original text of the questions that included all the answer choices. This was done by removing the incorrect answers, which thereby resulted in an improvement in performance. This is similar to what was seen in the previous edition of the task. Taking the TFIDF baseline with the basic pre-processing we got a MAP score of 0.3065 on the hidden test set. Taking this as the starting point, we built a $SentenceBERT$ Model in which we converted all questions and explanations into contextual word embedding vectors and ranked the explanations in descending order of cosine similarity of the embedded vectors. We observed a drop in the model's performance with the MAP score of 0.2427 on the test dataset, which is worse than the simple TFIDF ranker. We realized that it was the semantic overlap between the question and the irrelevant explanations that caused such an unexpected performance drop on further inspection. So we noted that we should not use contextual word embeddings, but instead, we must improve the simple but

effective information retrieval technique of TFIDF for the ranker. We then used the Sublinear TFIDF[2] and Binary TFIDF. The optimized Sublinear TFIDF vectorizer gave a boost in the score: 0.3254 MAP.

4.2 Preprocessing

It was seen that the TFIDF algorithm was very sensitive to keywords, so we applied the pre-processing and optimization techniques mentioned in (Chia et al., 2019). For each of these, we performed Penn-Treebank tokenization, followed by lemmatization using the lemmatization files provided with the dataset.[3] We used NLTK for tokenization to reduce the vocabulary size needed by combining the different forms of the same keyword. We also removed stopwords, which thereby removed noise in the texts. A simple TFIDF based ranker along with the above pre-processing returned a MAP score of 0.3850. Substituting Sublinear TFIDF, we noticed that the score increased to 0.4080 MAP. With some experimentation, we were able to further improve the MAP score to 0.426 using Binary TFIDF. Finally, we applied Recursive TFIDF as proposed in this paper (Chia et al., 2019), in which the authors treated the TFIDF vector as a representation of the current chain of reasoning, each successive iteration built on the representation to accumulate a sequence of explanations. We optimized all the other variables like $normalization$, $maxlen$, $hops$, $scale$. We found the MAP to be completely independent of the normalization used. For $maxlen$ = {128, 125, 144}, we found $maxlen$ = 128 to be most efficient. For number of hops = {1,2,3}, we found 1 to be best. This may be because semantic drift creeps in as we explore the nodes further away from the current node. A scaling factor was used in each successive explanation as it is added to the TFIDF vector. For the downscaling factors= {1.25, 1.3, 1.35}, we found 1.25 was optimum. We got a slight improvement in the score 0.4430 MAP when used along with Binary TFIDF. All these steps were done as a part of the pre-processing step.

4.3 Pure Language Model approach

After doing all the pre-processing steps, we tried to apply a simple pure language model based approach that has shown good performance in Text Classification tasks. We took each processed question and concatenated each of the 9029 explanations to it one by one. Then for each of these question + explanation pairs, we used a simple language model based BERT classifier ($BERTForSequenceClassification$) to predict whether the explanation was one of the gold explanations for that question. The result for this was 0.4116 MAP, which was lesser than we expected. We deduced that there are two major problems with this simplistic approach.

- **Class imbalance:** Most of the question-explanations would be labeled 0 (False), since out of the 9029 total explanations only a few would actually be gold explanations for the question. This causes the classifier to output the 0 label almost all the time and prevents it from learning the true relations between the gold explanations and the question. It is possible that the class imbalance could be mitigated by searching for better hyperparameter values; however, we weren't able to do that with the available resources, so we applied a different technique to address this.
- **Non-scalability:** This approach would require inferences equal to the number of explanations in the corpus for every question. While it's possible to do this for this relatively small corpus of 9029 explanations, as the number of explanations becomes larger, this approach would no longer be feasible; requiring too much time for training and, more importantly, for inference.

4.4 Using TFIDF to retrieve relevant explanations

To address the above problems, we applied the optimal TFIDF vectorizer (TFIDF_binary + recursive) obtained in the pre-processing step to first obtain the most relevant explanations for a given question based on the lexical overlap between the question and the explanations. The number of explanations retrieved by this initial ranker (top_k) was a tuned parameter. This technique was very effective at retrieving the gold explanations for a question. We have shown the fraction of gold explanations retrieved when we

[2] https://nlp.stanford.edu/IR-book/html/htmledition/sublinear-tf-scaling-1.html
[3] PTB tokenization and stopwords from the NLTK package

Method	MAP Score
TFIDF + preproc	0.3850
sublinear + preproc	0.4080
binary + preproc	0.4267
recursive	0.4429
sublinear + recursive	0.4429
binary + recursive	0.4430

Table 4: Inital scores using only pre-processing and TFIDF

top_k	Fraction Retrieved
10	0.484
30	0.676
50	0.767
80	0.837
100	0.865
300	0.965
500	0.987

Table 5: Value of parameter top_k vs Fraction of gold explanations retrieved

consider the top_k explanations in Table 5. We can see that almost 87% of the gold explanations are retrieved when we that top 100 explanations from TFIDF, and almost 99% of gold explanations are retrieved when we took the top 500 explanations. This saves our computation as we now need to only train the model for at max 500 explanations per question instead of 9029 explanations. Now top_k retrieved explanations are concatenated to the questions as in the previous approach, and the classifier model is trained to classify whether a given explanation among the top_k explanations is the right explanation for the question or not. The MAP score using the $BERTForSequenceClassification$ model was 0.4365 MAP. We inferred that the score was low because the model was predicting the 0 label for almost all inputs since there was still a significant imbalance in the training dataset (though significantly less than before).

4.5 Addressing class imbalance

To address the class imbalance in question explanation pairs, we applied a simple approach of oversampling of the minority class (Positive or '1' label). We simply repeated the gold explanations during training such that for each question, the number of positive and negative labeled explanations would be equal (equal to top_k/2). Hence the explanations for a given question were the top_k/2 negatively labeled explanations plus the positively labeled explanations retrieved by TFIDF repeated top_k/2 times. This was only applied while training, not during inference in the validation and test datasets. Using this simple technique, we were able to get a significant boost in the performance for the baseline of BERT with 0.4506 MAP score.

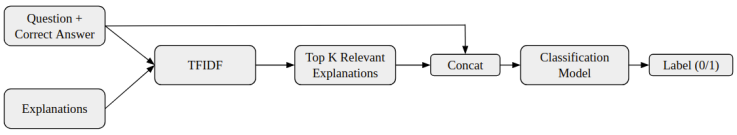

Figure 1: The Overall Flow of the Model

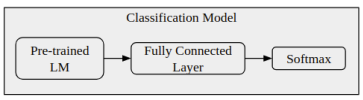

Figure 2: Inside the Classification Model

4.6 Pre-trained Language Models

We tried out all pre-trained language models available for sequence classification. We optimized the following hyperparameters: top_k, num_train_epochs, $batch\_size$, $learning\_rate$, $epsilon$, $gradient\_accumulation\_steps$, max_grad_norm, $weight\_decay$. The $batch\_size$ is dependent on the GPU RAM available. The parameters top_k and num_train_epochs are a function of training time. Since we needed to optimize the training time, we first trained all models with top_k as 100 with 3 epochs to get a preliminary model performance. Then we took the best models and trained it for a higher top_k (500 or 300 whichever was feasible) to get a boost in score. Our best performing model took close to 8 hours to complete the training. Further details of the models are given in the supplementary.

Parameter	Value
Optimizer	AdamW
Learning Rate	2e-5
Epsilon	1e-8
Max Grad Norm	1
Gradient Accumulation Steps	1

Table 6: Hyper Parameter Values

5 Results and discussion

We present the scores in the table given below. We got our best performance from RoBERTa. When we observe the results, we see that there is only a slight variation in the final scores of most pre-trained language models. We observe that the models overfit the given data. We could not perform a grid search to optimize all parameters due to computational constraints and had to manually search for the best hyperparameters due to which the performance of any given model may not be optimal. Further, we have trained only RoBERTa and BART for top_k 500 explanations and not other models because they had a long training time or a higher RAM requirement.

Model	Num Param	Batch Size	top_k	Train MAP	Dev MAP	Test MAP
RoBERTa (optimized)	355M	256	500	0.7210	0.5184	0.5061
RoBERTa	**355M**	**256**	**500**	**0.6708**	**0.5062**	**0.4902**
RoBERTa	355M	256	100	0.6182	0.4800	0.4798
BART	406M	128	300	0.7167	0.5036	0.4865
BART	406M	32	100	0.6708	0.4670	0.4769
SciBERT	110M	256	100	0.6544	0.4950	0.4855
ELECTRA	355M	128	100	0.6143	0.4943	0.4854
ALBERT	223M	32	100	0.6280	0.4813	0.4731
DistilBERT	134M	256	100	0.6049	0.4793	0.4641
BERT	110M	256	150	0.5776	0.4609	0.4506

Table 7: Final Results

6 Conclusion

We have given a system description of our team ChiSquareX which stood 4th place in the evaluation phase leaderboard with a MAP score of 0.4902. We have presented a system with optimized pre-processing of the dataset followed by an optimized TFIDF information retrieval scheme to obtain initial ranks, and then further pre-trained language model based re-ranker to rank the final explanations. Despite the computational constraints, just by leveraging Google Colab and other open-source tools, we have managed to fine-tune state-of-the-art pre-trained language models like RoBERTa, BART and ELECTRA on the (Xie et al., 2020) dataset and achieve a reasonable MAP score.

Acknowledgements

We would firstly like to thank the organizers Peter Jansen and Dmitry Ustalov for holding this shared task. It was a great learning experience for us. We would also like to thank the participants of TextGraphs-2019; their work was a great source of inspiration for us as to how to proceed with the task. (Chia et al., 2019) in particular, was a source of a number of simple but effective text pre-processing techniques to greatly improve performance. Additionally, we would like to extend a big thanks to the makers and maintainers of the excellent HuggingFace (Wolf et al., 2020) repository, without which most of our research would have been impossible.

References

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. Scibert: A pretrained language model for scientific text. In *EMNLP/IJCNLP*.

Yew Ken Chia, Sam Witteveen, and Martin Andrews. 2019. Red dragon AI at TextGraphs 2019 shared task: Language model assisted explanation generation. In *Proceedings of the Thirteenth Workshop on Graph-Based Methods for Natural Language Processing (TextGraphs-13)*, pages 85–89, Hong Kong, November. Association for Computational Linguistics.

Ondřej Dušek, Jekaterina Novikova, and Verena Rieser. 2020. Evaluating the State-of-the-Art of End-to-End Natural Language Generation: The E2E NLG Challenge. *Computer Speech & Language*, 59:123–156, January.

Daniel Fried, Peter Jansen, Gustave Hahn-Powell, Mihai Surdeanu, and Peter Clark. 2015. Higher-order lexical semantic models for non-factoid answer reranking. *Transactions of the Association for Computational Linguistics*, 3(0):197–210.

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, Hong Kong. Association for Computational Linguistics.

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? In *Proceedings of the Twelfth Workshop on Graph-Based Methods for Natural Language Processing (TextGraphs-12)*, pages 12–17, New Orleans, Louisiana, USA, June. Association for Computational Linguistics.

Heeyoung Kwon, Harsh Trivedi, Peter Jansen, Mihai Surdeanu, and Niranjan Balasubramanian. 2018. Controlling information aggregation for complex question answering. In Gabriella Pasi, Benjamin Piwowarski, Leif Azzopardi, and Allan Hanbury, editors, *Advances in Information Retrieval*, pages 750–757, Cham. Springer International Publishing.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online, July. Association for Computational Linguistics.

Y. Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, M. Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *ArXiv*, abs/1907.11692.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Huggingface's transformers: State-of-the-art natural language processing.

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.