Analyzing Causal Relations in Socio-political Events Extracted from Text: A Comparative Study of Large Language Models and Pretrained Language Models

Arsalene Khachmadi[1]

[1]*EURECOM*

Supervisor: Youssra Rebboud, Pasquale Lisena, Raphael Troncy

Abstract—This report explores the application of Large Language Models (LLMs) such as Zephyr-7B for data augmentation and pretrained models such as T5 and RoBerta for extracting causal relations from socio-political texts. We utilize LLMs to generate causal sequences, highlighting specific strengths such as the generation of varied and complex examples that enhance the training dataset. On the other hand, we employ the T5 model to focus on the extraction of these causal relations, demonstrating its proficiency in identifying and interpreting causality within the textual data. This study not only advances our understanding of NLP capabilities in handling causal analysis but also points to future research directions to further refine these technologies.

Keywords—*Causal relations, Large Language Models (LLMs), Pretrained Language Models (PLMs), NLP*

Contents

1. Introduction

The automatic extraction and analysis of causal relations from textual data have become pivotal in enhancing the capabilities of natural language processing (NLP) applications, ranging from enhancing semantic search engines to automating content summarization and supporting decision-making processes in socio-political contexts. This study focuses on a critical aspect of text analysis: identifying and understanding causal relationships within socio-political events described in various text sources.

For instance, consider the sentence: "The economic sanctions imposed by the United Nations caused a significant decline in the country's GDP, leading to widespread public unrest." This example illustrates the causal chain where the sanctions are the cause, the decline in GDP is the effect, and the subsequent public unrest is a further consequence.

Our research stems from the challenge posed by the inherent complexity of language and the subtle nuances that define causal relationships. Traditional language models, while proficient in many NLP tasks, often struggle to discern the underlying causal structures that dictate the logical flow of events within text. This limitation motivates the need for advanced methodologies that not only detect causal triggers and markers but also accurately interpret the directionality and significance of these relationships.

In this project, we aim to bridge this gap by implementing and exploring two distinct approaches: leveraging the robustness of Large Language Models (LLMs) like Zephyr-7b, GPT2/3 and the specialized capabilities of pretrained language models such as RoBerta and T5. Our goal is to evaluate and enhance the effectiveness of these models in parsing complex narrative constructs to extract causal links, thereby providing deeper insights into how events are interlinked in socio-political narratives.

Through systematic experimentation and comparative analysis, this study will not only contribute to the academic and practical understanding of causal relation extraction but also aims to refine the methodologies used in this domain, setting a benchmark for future research endeavors in advanced text analysis.

2. State of the Art

2.1. Overview of Boshai's Work

Boshai's[3] work, presented at the RANLP 2023 Shared Task, focuses on the automatic extraction and analysis of causal relations from textual data. The primary goal of this research is to improve the accuracy and efficiency of detecting causal relationships within complex narrative structures. Boshai utilizes advanced natural language processing techniques to develop models that can identify and interpret causal links between events described in texts. This work is crucial in enhancing the understanding of how events are interconnected, particularly in socio-political contexts. It has two subtasks:

- **Subtask1** a binary classification task that deals with the classification of sentences into causal and non-causal. A sentence is labeled causal if it contains any cause-effect chain.
- **Subtask2** is about extracting the exact token spans of Cause (<ARG0>), Effect (<ARG1>), and Signal (<SIG0>). There are up to four different causal relations within a single sentence.

2.2. Technical Details

BoschAI's method achieves state-of-the-art performance in causal relation extraction by combining custom algorithms with pretrained language models. Models like BERT [1] and RoBERTa [2] are fine-tuned using specially annotated datasets used in [3] that emphasize textual causal triggers and effects. A key feature is their multi-layer neural network architecture with attention mechanisms to capture subtle links between words and sentences.

BoschAI used RoBERTa-Large and BERT-Large for embeddings, processed through a linear layer and tagged using the BILOU scheme. A Conditional Random Field (CRF) layer computed the most consistent tag sequence, optimized with AdamW. Metrics like precision, recall, and F1-score demonstrate notable improvements in recognizing and comprehending causal relationships compared to baseline methods.

In addition to BoschAI's method, we used another method. The goal of Idiap's [7] CNC Shared Task was to extract causal linkages utilizing a variety of cutting-edge methods. They used models like T5 for causal relation extraction in their research. T5 was adjusted using annotated datasets created especially for determining causal links. By using this method, the model was able to produce embeddings that

faithfully depicted the text's causal relationships. Idiap's research endeavors were designed to improve the comprehension and efficacy of pretrained language models in identifying causal relationships, hence offering significant perspectives and techniques in the domain.

3. Theoretical analysis

3.1. Architecture

In the section, we will show the difference between the architecture of Bert/RoBerta and T5.

BERT/RoBERTa: Both are Transformer-based models focused on bidirectional context, utilizing attention mechanisms to understand relationships between words. RoBERTa is a refined version of BERT with more training data and optimized hyperparameters. The figure 1 below shows the architecture for the RoBerta.

T5: Also Transformer-based, T5 is designed for text-to-text tasks, treating every NLP problem as a text generation task, enabling a more flexible and generalized approach. Figure 2 shows the T5 architecture.

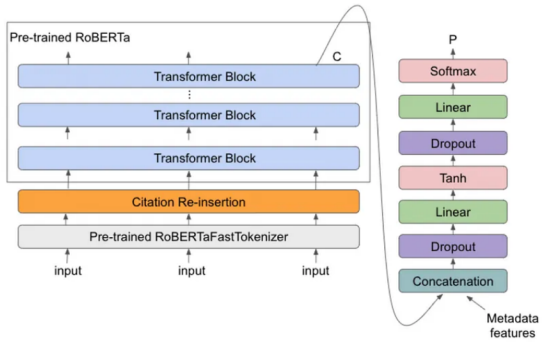

Figure 1. RoBerta model

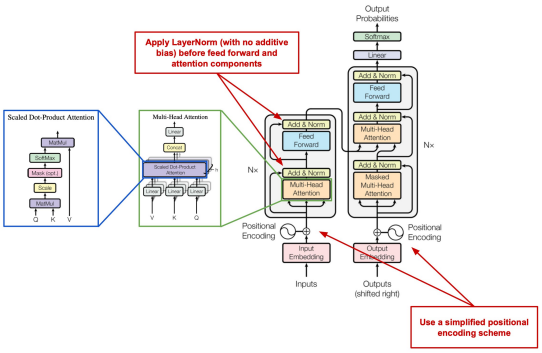

Figure 2. T5 model

4. Data exploration

4.1. Data analysis

For a starter, we had three datasets:

Train set: it has mainly a text column and text with pairs column indicating the text containing tags or spans, which are three, cause, effect and signal, and here's an example of a sentence in the "textwpairs" column: "<ARG0>The LMC workers later protested</ARG0><SIG0>by</SIG0><ARG1>dumping garbage on the Hardoi road and blocking traffic</ARG1> "

Validation set: basically the same as the training set.

Test set: which is unlabeled.

4.2. Data augmentation

For the improvement of data quality and future results, we decided to do data augmentation with the method prompt engineering. The goal is to generate new data and add it to the old one. This task involved using three models, GPT2 as a start, but we observed that the quality of the generated text was not good enough and the size of the data was not as expected, so we decided to do further trials to use, at the end, Zephyr-7B [6] which is a fine-tuned version of Mistral-7B. Since Zephyr-7B is a complex model, we used a quantified version to reduce memory requirements, AutoAWQ [4]. Since quantified versions always need updates, at the end we were not able to run the model, so we finally used microsoft/Phi-3 [5], but unfortunately we had poor result with the last one, so we did collect the generated text from the experience with [6].

Below is a snippet that was used to generate the text:

```
domains = ['economic events', 'social events', 'political unrest', 'elections', 'natural disasters',

request = """
generate me 5 causal examples for each domain in domains
"""

definition = """Below are the definitions for the tags in the sentence:
  Cause: The reason for an event happening, to be enclosed between <ARG0> and </ARG0>.
  Effect: The event that occurs due to the cause, to be enclosed between <ARG1> and </ARG1>.
  Signal: Words that transition the cause to the effect, to be enclosed between <SIG0> and </SIG0>.

  Please generate causal sentences within this domains."""
```

Figure 3. Prompt engineering

Even if Zephyr-7B was unable to produce multi-causal element sentences that followed the CNC standard, this is an interesting direction for further investigation. The process involved utilizing a higher GPU due to the complexity of the model, with prompts created in a Jupyter Notebook environment. Processing was done on the CSV file containing the output that Zephyr-7B produced in answer to the prompts. In order to ascertain whether employing Zephyr-7B to generate augmented datasets was worthwhile, a tiny section of the dataset was manually assessed in a previous phase of the study. It was discovered that at least 90% of the 20 randomly selected entries were valid after evaluation.

4.3. Data preprocessing and transformation

In this section, we saved the generated text into text file, and then processed the data to extract domain-specific causal text pairs.

To store the extracted information, we initialized lists for domains and causal text pairs, along with variables to hold the current domain and its associated pairs. As we processed each line of the file, we identified the end of a domain section by checking for lines starting with </s>. When such a line was encountered, the current domain and its pairs were added to their respective lists. If a line indicated a new domain by starting with in the domain of, the domain name was extracted and stored. Any other lines were considered part of the causal text pairs and were collected accordingly.

After processing all lines, we ensured that the last domain and its pairs were also added to the lists. These lists were then used to create a pandas DataFrame, organizing the domains and causal text pairs into columns. Finally, the DataFrame was saved as a CSV file with the columns arranged in a specified order, and a confirmation message was printed indicating where the processed data had been saved. This sequence of actions effectively transformed the structured text data into a tabular format suitable for further analysis or reporting.

5. Experimental results

5.1. Results with Roberta

In this section, we duplicated the previous work, and we got approximately the same results as the previous experience. However we used different versions of Roberta and Bert, to reduce the use memory and we also needed to reduce the hyper-parameters such as number of epochs and batch size, we summarized the results in the figure 4 and 5 below:

	Precision	Recall	F1 score	Accuracy
Roberta-large(original model)	0.84	0.64	0.701	
Roberta-base	0.85051546	0.89189189	0.8707124	0.86
Bert-base	0.83425414	0.81621622	0.82513661	0.81

Figure 4. Results

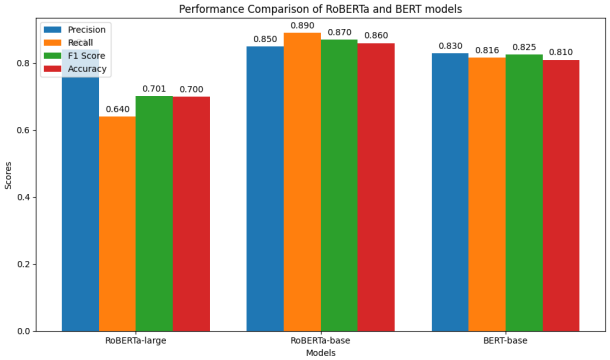

Figure 5. comparison between the models

As we can observe, the results of the Roberta-base model outperform the other two models in this comparison. The reason behind this is the fact that it strikes a compromise between efficiency and complexity. By adding more encoders and attention heads to the original BERT architecture, RoBERTa-base and RoBERTa-large both improve it. However, the base model performs better on smaller datasets since it trains more quickly and has a lower chance of overfitting. In practical circumstances, RoBERTa-base often performs better than its larger equivalent, which could be explained by the improved generalization that results from this balancing, thereby achieving higher accuracy, precision, recall, and f1-scores compared to the other models.

The reason behind the results of Roberta-large are justified by the fact that larger models require significantly more computational resources and time to train effectively. And since Roberta-large was not given sufficient training time or computational power (we training only for 7 epochs), it might not have reached its full potential, resulting in suboptimal performance compared to Roberta-base.

5.2. Results with T5

The T5 model was a crucial part of our project due to its complexity and innovative architecture, which frames all NLP tasks as text-to-text problems. This versatility makes T5 highly effective but also highly demanding in terms of computational resources.

High GPU Requirements

Large Parameter Count: T5 models have billions of parameters, requiring high-end GPUs with substantial memory to handle their size and prevent memory overflow.

Parallel Processing: The self-attention mechanism in T5 involves computationally intensive operations that benefit significantly from GPU acceleration.

Computational Time

Extensive Training: Training or fine-tuning T5 on large datasets over many epochs takes a lot of time.

Hyperparameter Tuning: Finding the optimal hyperparameters involves extensive experiments, each consuming significant computational power.

Inference Speed: Even making predictions with T5 can be slower due to its complexity, making powerful GPUs essential for efficiency.

The performance of T5 compared to Roberta-base did not meet our expectations. This discrepancy can be attributed to the model occasionally struggling to accurately predict all tokens in the input text. Unlike token tagging, which only processes the source text, the seq2seq nature of the T5 model can lead to token mix-ups or omissions.

In the figure below, we analyze the model's performance to determine if it is training effectively. The model shows improvement over time, approaching a minimum evaluation loss value of 0.072, indicating convergence. The gradient norm over epochs displays an overall increasing trend, suggesting significant updates to the model's parameters. However, large spikes may indicate potential instability in the optimization process.

The tables below present the results for both metrics from the original dataset and the augmented data:

Metric	Overall	Cause	Effect	Signal
Recall	0.536	0.506	0.470	0.688
Precision	0.549	0.506	0.470	0.738
F1	0.540	0.506	0.470	0.702
Accuracy	0.490	-	-	-
Number	409	249	249	160

Table 1. Model Performance Metrics

Metric	Overall	Cause	Effect	Signal
Recall	0.532	0.506	0.470	0.669
Precision	0.540	0.506	0.470	0.703
F1	0.534	0.506	0.470	0.678
Accuracy	0.494	-	-	-
Number	409	249	249	160

Table 2. Model Performance Metrics with Augmented Data

And below are the figures showing the results for both training and evaluation losses

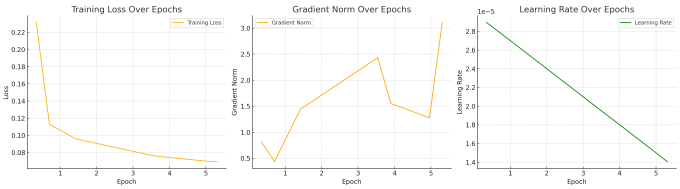

Figure 6. Training loss over epochs

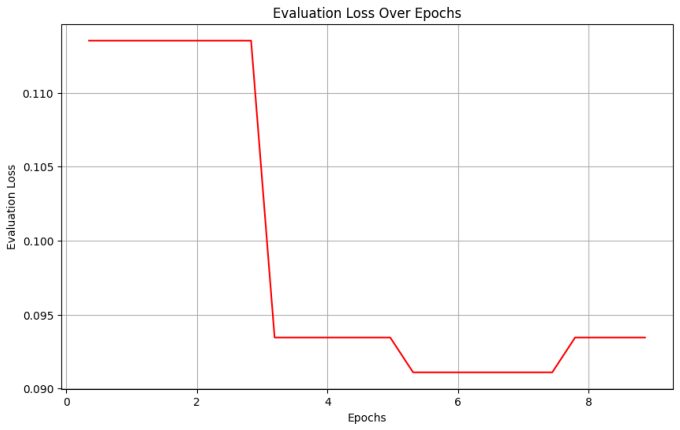

Figure 7. Evaluation loss over epochs

5.3. Comparaison

Comparing the results of the original and augmented datasets, the performance metrics indicate that data augmentation did not significantly improve the model's performance. For overall metrics, there

is a slight decrease in recall (from 0.5365 to 0.5319), precision (from 0.5487 to 0.5403), and F1 score (from 0.5400 to 0.5341) with the augmented data. Accuracy remains nearly the same, showing a minimal increase from 0.4904 to 0.4937.

For cause-related metrics, recall, precision, and F1 score remained unchanged at 0.5060 with both datasets.

Effect-related metrics also stayed consistent across both datasets, with recall, precision, and F1 score remaining at 0.4699.

Signal-related metrics showed a slight decrease with the augmented data, with recall dropping from 0.6875 to 0.6688, precision from 0.7379 to 0.7033, and F1 score from 0.7019 to 0.6778.

These results suggest that data augmentation did not enhance the model's performance as expected. Instead, the original dataset yielded slightly better or equivalent results in most categories, indicating that the augmentation may not have provided additional beneficial information for the model to learn from.

6. Conclusion and Future Work

In this project, we used models such as BERT, RoBERTa, and T5 to investigate the extraction of causal links from socio-political literature. Our main conclusions demonstrated that RoBERTa-base performed better than other models because of its strong pre-training, which improved F1 scores, accuracy, precision, and recall. On the other hand, because of limited training time and computational resources, RoBERTa-large fared poorly. Although the T5 model demonstrated efficacy, it was beset by issues related to computing needs and infrequent overfitting, underscoring the necessity for additional tuning.

For future work we can significantly enhance the performance of the T5 model and address its current limitations. Superior outcomes in a variety of natural language processing tasks can be attained by refining the T5 model through improved prompt engineering, sophisticated training methods, and fine-tuning, as well as by utilizing high-performance computing and collaborative research.

Furthermore, training can be sped up and more models and datasets can be handled effectively by utilizing distributed computing power and high-end GPUs. Innovative methods and insights will be brought to further enhance causal relation extraction in NLP through interaction with the research community and cross-disciplinary collaboration. These aspects can be addressed to greatly enhance these models' performance in a variety of NLP tasks.

References

[1] Hugging-Face. "Bert." (), [Online]. Available: https://huggingface.co/docs/transformers/model_doc/bert.

[2] Hugging-Face. "Roberta." (), [Online]. Available: https://huggingface.co/docs/transformers/model_doc/roberta.

[3] boshai. "Boschai-cnc-shared-task-ranlp2023." (n.d.), [Online]. Available: https://github.com/boschresearch/boschai-cnc-shared-task-ranlp2023/tree/master.

[4] casper-hansen. "Autoawq." (n.d.), [Online]. Available: https://github.com/casper-hansen/AutoAWQ.

[5] Hugging-Face. "Microsoft/phi-3-vision-128k-instruc." (n.d.), [Online]. Available: https://huggingface.co/microsoft/Phi-3-vision-128k-instruct.

[6] Hugging-face. "Zephyr 7b-beta." (n.d.), [Online]. Available: https://huggingface.co/HuggingFaceH4/zephyr-7b-beta.

[7] idiap. "Cncsharedtask." (n.d.), [Online]. Available: https://github.com/idiap/cncsharedtask/tree/main.