Cross-Document Event Co-referencing using RoBERTa
Report and Analysis
By: Sohini Roy

Task

The task at hand is to fine-tune a small language model (RoBERTa) to improve performance on a cross-document event coreference task. The goal is to enable the model to correctly predict whether two event mentions in a sentence pair refer to the same real-world event.

Methodology

We began with preprocessing. In the train, dev, and test data, we were able to separate out the sentences, event triggers, and labels the same way, due to the formatting being consistent. Each line was tab-separated, so after splitting the sentences into tokens based on tabbing, we accordingly added sentences and triggers to their individual lists (i.e. first sentences in one list, second sentences in another, and same for the triggers), and all labels in one list. The tokenizer was created by passing in each sentence list as an individual parameter.

I used a PEFT (Parameter-Efficient Fine-Tuning) model with a LoRA (Low-Rank Adaptation) config in order to fine-tune only a small subset of parameters/add lightweight adapters, to reduce memory usage and training time.

Results

Upon my first evaluation of the dev dataset, I had the following hyperparameters:

- Batch size per device (train) = 8,
- Batch size per device (dev) = 8,
- Number of train epochs = 3,
- Learning rate = 5e-5,
- Weight decay = 0.01

These resulted in the following metrics:

- Accuracy: 90.34%
- Precision: 42.27%
- Recall: 21.1%
- F1: 28.15%

Next, to experiment, I slightly increased the learning rate to 1e-4.

That resulted in the following metrics:

- Accuracy: 93.17%
- Precision: 61.9%
- Recall: 54.48%
- F1: 57.95%

These metrics are quite a bit better than our initial findings, specifically the ones for precision, recall, and F1. If I had the resources and time, I would try to play around with tuning some more hyperparameters, such as number of epochs or batch size.

Now, we try running the evaluation metrics on the test dataset with the updated learning rate. That resulted in the following metrics:

- Accuracy: 91.36%
- Precision: 0.0%
- Recall: 0.0%
- F1: 0.0%

This is obviously not ideal, but given the tricky dataset we have been given to work with, it actually makes sense. There is a huge imbalance in the data i.e., of 220k examples, only 10k are positive. My guess would be that it is definitely predicting the wrong class for almost all of the examples, not to mention that the test dataset contains a lot (if not all) of information about the same event repeatedly, so this could also be skewing it.

I would try to clean up the data imbalance and see what effect this would have on the modeling. I would also like to tune the hyperparameters further- learning rate seemed to affect RoBERTa quite positively, so maybe even increasing that a bit more would be helpful.

Future Work

My future work on this project would consist of:

- Downsizing and balancing the data
- Experimenting further with hyperparameters, as well as different GPU runtimes to see which one is the best. I ended up using A100 which seemed to work for me better than T4.