Report

1. Dataset

1.1 Selecting dataset

First, we need a dataset. After searching for existing ones, I found GYAFC dataset which seemed really suitable for this task, but it was not available. There were a few others that were either not available or not really suitable.

Eventually, I picked `pavlick-formality-scores` dataset, which contains sentences from news, blogs, email, and QA forums. All of them are labeled with a number from -3 to 3, indicating how formal the sentence is, from lowest to highest.

1.2 Transforming data

This is great, but I think that binary classification is more suitable here, firstly we don't want to overcomplicate the task, but also I think that it doesn't make much sense to differentiate phrases that have labels -3 and -2 for example. They are clearly both informal, and moreover this labels are kinda subjective, so giving the exact numerical answer is a little odd here. That's why I decided to use binary classification. So now we need to somehow label this sentences just -1 and 1

First idea is kinda stupid, just say that all positive values belong to positive class, and other belong to the negative. I don't really like it, since values around zero, or even equal to zero shouldn't belong to either positive or negative class, they are almost neutral. And since dataset is not too small we can afford to drop the values that are within some threshold around 0, let's say $[-0.5, 0.5]$

2. Metrics

We are dealing with binary classification so there are multiple good metrics existing, like Accuracy, Precision, Recall, F1, AUC-ROC. I think here a good option would be F1, Recall and Precision. In this problem we might have some phrases that are ambiguous and it would be better to put them into formal class, because doing otherwise might not make sense. For example it's better to label incorrectly some neutral polite phrase like "Hello, do you need any help?" as formal, than informal. That is why we want to focus on the amount of False positive and False negative predictions and analyze how the model behaves with ambiguous sentences

3. Formality detection approach

3.1 RoBERTa base

Time to pick the formality detection approach, I want to use some fine-tuned model, I searched on hugging face, and there were a lot of different options, I know that my laptop is really slow so I wanted to use some lighter model, therefore I decided to pick `roberta-base-formality-ranker`

Unfortunately, I don't have GPU so it was quite long to wait for the end of execution. I was using Jupyter Notebook before, so at this point I switched to Google Colab to be able to use GPU, and of course it worked much faster

3.2 DeBERTa large

Also I wanted to use a bigger model, in order to compare it with the previous one, so I chose `deberta-large-formality-ranker`.

4. Conclusion

Regarding the first model, I got that all metrics were high, the least one is precision with a value 0.8342, so model has quite many false positive predictions, and combining this with the fact that recall is 0.9699, meaning that we had not that many false negative predictions, I can suppose that model is a little bit biased towards predicting positive class. Just to make sure that the issue is not in dataset, I checked that classes are balanced, which they were

Looking at the results of the larger model, they were significantly lower for some reason, the only thing that was better is recall with a value 0.9719. And with precision being only 0.7125, the bias towards positive class is even more noticeable here

Overall, the performance of both models on this dataset is not bad, but first model did a much better job. And I think that the fact that both models tend to give more positive predictions, meaning predicting that sentence is formal, is not a bad thing, because there are scenarios where it is beneficial, and doing otherwise would be misleading