CSI 5386
Natural Language Processing

Professor
Diana Inkpen

Final Project

Evaluation on Data Balancing Methods for Deep Learning using the COLIEE2021 Dataset

Submitted By
Michel Custeau 8658589
Beril Borali, 300036112

December 22, 2022
University of Ottawa

Introduction:

When dealing with artificial intelligence applied to the field of law, one of the biggest challenges is the lack of data. Since the field of Legal AI is relatively new, there is not a lot of data available, and it doesn't help that, given the sensitivity around ethical questions such as privacy, some clients and organisations might be reluctant to share large amounts of legal data. This creates difficulties particularly with Deep Learning, which usually requires very large amounts of labelled and unlabelled data. Given this, we decided to evaluate different methods for dealing with data imbalance in a legal corpus. These three methods were leaving the data unchanged and training the models with less epochs, undersampling the negative class, and oversampling the positive class. Throughout this report we will discuss the methodology behind these techniques and whether or not these methods benefit deep learning models in the context of the data imbalance in the task 2 dataset of the COLIEE2021 competition.

Task Description:

In a basic sense, this task is to find whether or not two pairs of legal paragraphs entail each other. In more details, we are given legal documents where a court decision has already been located for us inside each document. Hence, while we are given entire documents, we only really care about the paragraph that contains the court decision for each. Alongside the given court decisions in the documents, for each, we are also given a list of paragraphs from a separate legal document, where our goal is to find which of these paragraphs entail the court decision.

Related Literature:

The task 2 dataset has been previously studied in other papers for information retrieval, such as Bert PLI in which the authors use the task 2 dataset from the 2019 competition to tune a model that they include in an information retrieval pipeline (Rosa et al., 2022). As for the specific dataset for the 2021 competition, the winners of the competition have released a paper on their results, however we found their results hard to replicate since some important details on how they dealt with class imbalance were left unclear (Shao et al., 2020). While we were unable to match their results, our goal for this paper is to add more clarity around what potential data balancing approaches could work for future experiments to try to match or beat the winners results in the 2021 competition and also future ones.

Dataset:

The training set contained 425 decisions each with a unique set of potential entailment paragraphs associated with them. The test set contained 100 decisions each also having a unique set of potential entailment paragraphs associated with them. When we paired up all the court decisions with their respective potential entailment paragraphs, this then amounted to:

Training Set			Test Set		
Positive Examples	Negative Examples	Total Examples	Positive Examples	Negative Examples	Total Examples
499	14171	15216	117	3407	3524

This meant that in total the training set contained 499 positive examples of entailment and 14717 negative examples, a big red flag as the negative class completely outnumbered the positive class, making models very prone to overfitting.

Models:

The three models used were T5-base and sBert with bert-base and sBert with Legal Bert for evaluation, and bm25 for baseline accuracy. T5 is a text-to-text transformer which contains both an encoder and a decoder. Hence it can receive text as input, and output text based on it. This is different from Bert which only has encoder blocks. For this task, we trained T5 to output “true” when entailment was found between the legal paragraph pair and “false” when no entailment was found.

sBert is a Bert model that creates sentence and paragraph level representation using mean polling on the final layer of token embeddings. Since sBert doesn't have a decoder like T5, and simply returns an embedding for each paragraph, we measure similarity using a threshold of 0.7 for cosine similarity between the two paragraphs. In other words, if the cosine similarity between the two embeddings it created was above 0.7, this was considered as entailment, and if it was under, then no entailment was found. As stated before, we used two pretrained models for sBert which were bert-base and legal bert, a bert model pre-trained on 12 GB of English legal data.

The reason we use bm25 as baseline accuracy is since its a non neural network model that doesn't require any learning, but instead calculates relevance using an equation similar to tf-idf but upgraded for also taking document length into account, instead of tf-idf which mainly just looks at words and their importance but doesn't take the length of document into account. Hence, bm25 is definitely advantaged over the neural network methods for this task, as when the data is heavily skewed, one often might opt to use a model that doesn't require training.

Evaluation:

For evaluation, we used F1 score, recall and precision. The reason we chose these metrics is that those are what the COLIEE2021 recommended to use. Furthermore, since we were mainly interested in correctly predicting the positive class instead of how well we could predict the negative class, these metrics suited our goal well.

Precision is calculated by the number relevant retrieved documents over the total number of retrieved documents.

$$\text{precision} = \frac{|\{\text{relevant documents}\} \cap \{\text{retrieved documents}\}|}{|\{\text{retrieved documents}\}|}$$

Recall is calculated by the number of relevant retrieved documents over the total number of relevant documents.

$$\text{recall} = \frac{|\{\text{relevant documents}\} \cap \{\text{retrieved documents}\}|}{|\{\text{relevant documents}\}|}$$

Finally, we then have the f1 score which is calculated by using both the precision and recall in this equation

$$F1\ Score = 2 \times \frac{recall \times precision}{recall + precision}$$

Methods:

As mentioned previously, the three data balancing methods used were leaving the data unchanged, undersampling the negative class to match the positive class, and oversampling the positive class to match the negative class.

When leaving the data unchanged, it was important not to train our models for a long amount of epochs, as overfitting created by the dominating class could quickly start to deteriorate our results. This meant that all models had to be trained on less epochs than normally for the unbalanced data.

For undersampling, the goal is to remove examples from the dominating class. This in our case meant to remove negative examples until the number of positive examples and negative examples were equal in number. Hence, the negative examples were randomly chosen to be removed until we had only 499 of them left, making both the positive and negative class equal to 499.

For oversampling, the goal instead is to create artificial examples of the minority class. In our case this meant creating paragraph pairs that entailed each other. To do this, instead of only looking at the court decision, we looked at the entire paragraph that contained the court decision. This was done by moving a sliding window through the paragraph, where each step of the sliding window was paired with the associated entailment paragraphs of the court decision. This created a new artificial positive pair for each step of the window. This then led to the question of what size of window to choose. If we chose a static window size, the model might overfit by recognizing the artificial examples from the real ones by their length, since it might learn the number of words these artificial examples always have. To get around this,

for each case, the window size was randomly selected in the range between 25 and 55 words. We felt that this range was appropriate since the average length of the decisions for the real positive examples was around 37.5. The stride for the window was its length divided by 9. The reason we chose this stride number was that it made the number of artificial examples almost the same as the number of negative examples. Since the size of the window for each case is chosen randomly, running artificial positive examples creation gives a different number of artificial positive examples every time. Hence, the number of positive examples used for T5 were 14199 and for the two sBert models it was 14439. This meant that in both cases the number of positive examples was now almost the same as the number of negative examples, which meant the data was balanced.

Parameters:

T5-base was trained with a batch size of 4 with an AdamW optimizer equipped with a learning rate of 3e-4. For the original unbalanced data, we trained for 5 epochs, for the oversampled data 13 epochs, and for the undersampled data we trained for 10 epochs.

Both sBert models were trained with a learning rate of 2e-05 and a batch size of 16. For the original unbalanced data they were trained for 3 epochs, for the oversampled data and undersampled data they were trained for 10 epochs.

The reason the original unbalanced data was trained on much less epochs than the oversampled and undersampled data for each model was due to the fact that after around 5 epochs the accuracy usually started to go down and would later crash to 0 as epochs would go on. The reason for crashing to 0 as more epochs went on was, because of the dominating

negative class overfitting the training, the model would simply predict the negative class for everything on the test set.

For the undersampled and oversampled class, it didn't really matter how many epochs you ran past the 10th since around the 10th epoch it appeared to stabilise to a certain value and not change drastically as the epochs kept going.

Results:

We will now showcase our results of our three models against the three methods.

Trained on unbalanced data			
Models	recall	precision	f1
T5-base	**0.435897436**	0.309090909	0.361702128
sBert: Bert-Base	0.41025641	0.436363636	0.422907489
sBert: Legal-Bert	0.427350427	**0.490196078**	**0.456621005**
Trained on oversampled data			
Models	recall	precision	f1
T5-base	0.341880342	0.32	0.330578512
sBert: Bert-Base	0.196581197	0.196581197	0.196581197
sBert: Legal-Bert	0.230769231	0.238938053	0.234782609
Trained on undersampled data			
Models	recall	precision	f1
T5-base	0.136752137	0.136752137	0.011196641
sBert: Bert-Base	0.264957265	0.074519231	0.116322702
sBert: Legal-Bert	0.025641026	0.01863354	0.021582734

As we can see, none of the oversampling and undersampling methods overall improved the results. Hence, this shows that less epochs on imbalance data performs better on this dataset than more epochs on balanced data through oversampling or undersampling. For T5, precision improved with training on oversampled data, however this badly affected recall and f1. Both sBert models did not benefit from the oversampling and undersampling in any way.

We can also see that Legal-Bert was our best performing model in terms of recall and f1, showing the importance and impact of pretraining on domain specific data.

If we now take our best performing model, we can now compare it to our baseline overall accuracy of bm25.

Models	recall	precision	f1
Legal-Bert	0.427350427	0.490196078	0.456621005
bm25	0.564102564	0.66	0.608294931

As we can see, our best results did not beat the non neural network model, hence showing that more improvements are needed for the deep learning models in this context.

Result Discussion, Future Work and Possible Improvements:

While these results show that just doing oversampling or just doing undersampling does not improve the results, we still believe that some type of data balancing could be of benefit to the deep learning models. Specifically, we believe that perhaps combining both undersampling and oversampling could be of benefit. The reason we believe this is that when we just do undersampling, the reason the overall accuracy drops is due to the fact that when we make the negative class around the same number as the positive class, there simply is not enough data left for the deep learning model to learn anything. But when we do oversampling, the reason the accuracy drops is because we are creating too many artificial examples in order to match the negative class which leads to too much noise and redundancy. However, if we were to reduce the number of negative examples to a point where they are less dominating, but also while keeping enough negative examples for the deep model to learn, while simultaneously creating enough artificial positive examples to match the number

of negative examples, but not too many to a point where they lead to too much noise, then we could potentially have good results, which we will experiment with in the future.

Specifically, the next steps could be experimenting with a larger stride (something bigger than the length of the window divided by 9), which in this case would move the window a larger distance in the paragraph at each step. The upside of this would be that it would lead to less noisy and more diversified positive artificial examples, the downside would be that it also would reduce the number of created artificial positive examples, which would not solve the initial problem of the dominating negative class. This is where undersampling a little bit the dominating negative class could potentially come into play to make up for less oversampling.

Another way we could improve on the results would be to train on COLIEE data from previous year competitions, or on the MS Marco dataset, which is part of what the past winners of the competition did. In our case, we did not try this since we wanted to focus more on improving the current data specifically with balancing, which as we have shown there is still more work that needs to be done on that front.

Conclusion:

In this report we have argued that simply doing oversampling to make the minority class match the dominating class, or doing undersampling to make the dominating class match the minority class does not lead to better results than simply training on unbalanced data for less epochs. We have also shown that while training unbalanced data on less epochs led to better results than the models trained on undersampled and oversampled data, these results are still not adequate as they are not able to surpass baseline bm25 results. Hence, to deal with this,

we have discussed why the data balancing did not work and how these flaws could potentially be fixed with future experiments where we combine the different data balancing techniques together.

References

Rosa, G. M., Rodrigues, R. C., Lotufo, R. de A., & Nogueira, R. (2022, February 7). *To tune or not to tune? Zero-shot models for legal case entailment*. arXiv.org. Retrieved December 22, 2022, from https://arxiv.org/abs/2202.03120

Shao, Y., Mao, J., Liu, Y., Ma, W., Satoh, K., Zhang, M., & Ma, S. (2020, July 9). *Bert-PLI: Modeling paragraph-level interactions for legal case retrieval*. IJCAI. Retrieved December 22, 2022, from https://www.ijcai.org/Proceedings/2020/484