MLiP 24 - Coleridge Initiative: Show US the Data

Bono IJpelaar (s1063058) Matthijs Neutelings (s4804902) Saskia de Wit (s1060762)

June 14, 2021

Introduction

In this report we will discuss the second project for the course Machine Learning in Practice. This second challenge is the Kaggle competition "Coleridge Initiative - Show US the Data"[1]. The goal of this competition is to predict which dataset(s) are used in a collection of papers. Based on a relatively small test set that contains scientific papers with their datasets labeled we will train a NER solution to try to predict the datasets of the papers in a hidden test set.

The code of our train script and our validation script can be found on our Kaggle.

Method

To start of the project we investigated the already existing notebooks in the code section of the challenge. In these notebooks we found a baseline notebook[2] on which we started working. This notebook had a literal matching approach, meaning that this notebook collected all the known labels of datasets that are present in the given training data. After collecting these labels it went through every paper in the test data, which only consists of four papers. For every paper it scanned if any of the known labels were present in the paper and if it found one that would mean that the dataset corresponding to the paper was found.

This approach worked for a fair portion of the hidden test set, which in first instance gave a score of around 0.7. After a later correction of the competition in which the hidden test data was altered this still gave a score of around 0.5. This would mean that about half of these papers contain datasets with labels that are already known from the training set. To improve our score we had to find a way to predict the unknown datasets.

To get this to work we searched for and found a new notebook that used BERT Named Entity Recognition. The aim for this approach is to train the BERT model to find dataset names based on the sentence structure in the papers.

BERT for Named Entity Recognition

In the notebook we found, Bidirectional Encoder Representations from Transformers (BERT) [3] was used for the task of finding the labels. BERT has no unidirectionality constraint (reading a sentence from left to right) because it uses a masked language model pre-training objective. This model randomly masks some of the input tokens and the objective is to predict the vocabulary id of the masked word. This is done based on its context. In the BERT framework there are two steps: the pre-training and the fine-tuning. The pre-training procedure follows the existing literature on language model pre-training. After that, fine-tuning is straightforward since BERT allows many downstream tasks by swapping out the appropriate inputs and outputs.

On Kaggle, two notebooks were used for these tasks. One model for the fine-tuning and one for pre-training. In the pre-train notebook, BERT was pre-trained using all data taking only the sentences that contained the words 'data', 'study' or the label itself. Also in the notebook in which the predictions were made, only the sentences containing 'data' or 'study' were used for prediction. This is done because only 10% of the sentences contains these words, but these sentences cover about 70% of the datasets. This takes time off predicting, as we only look at 10% of the sentences.

The baseline model was used for the first predictions and after that several approaches were taken for the pre-training and fine-tuning of the BERT model.

Training the BERT model

For pre-training the BERT model in different ways we used several methods. Two of those methods we based on our findings of analyzing the data that was available to us. Another we found through paper research.

Capitalized words We saw that in general datasets that were mentioned in the text consisted of sentences with multiple capitalized words, or when they were abbreviations, multiple capital letters in a single word. This lead us to implementing a script that pre-searches for those characteristics in the scientific papers sentences, because there was a higher likeliness of a dataset mention in such a sentence.

Dataset frequency Another method we tried after analyzing the data was equalizing the mentions of datasets. In the train set there were a total of 45 different datasets mentioned. However, the distribution of these mentions over the scientific papers was far from even. There were datasets that were mentioned several thousand times (ADNI) and datasets that were only mentioned once. For this we implemented a script that evened these mentions out. We took the average over all papers and sorted the datasets by number of appearances. Then we replaced dataset names with the highest amount of mentions with datasets that had the lowest amount of mentions, until all of them evened out.

Data Augmentation A final method we applied to try and improve the performance of our BERT model is doing data augmentation. In a paper we found we were introduced with this method for NER solutions[4]. We applied one of the methods proposed in the paper which is finding synonyms for the known dataset names. In this way we created a greater diversity in names of the datasets without changing the context of the dataset name too much. To apply this method for every word in a dataset name we replaced it with a synonym and in this way we created more variety in the dataset names to train our BERT model on.

Another method implemented to augment our data is a translation approach[5]. With this approach one translates the name of the datasets to a foreign language and then back to the original language. An example of this can be seen in Figure 1. This causes a slight differentiation in the names and thus giving a greater diversity of labels to use.

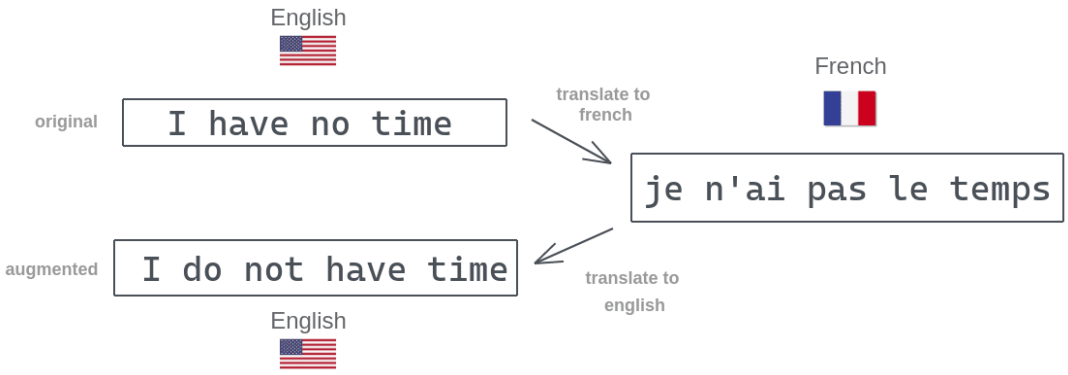

Figure 1: Data Augmentation through Translation [5]

Validation To validate the different approaches without having to submit to Kaggle every time, we created a validation set ourselves. We took the training data and checked which dataset labels were present, after doing this we excluded one of the labels, being "Our World in Data COVID-19 dataset", from the train set. After training the BERT model through one of our methods we ran the model on the validation set we made. This way we could check how the BERT model performed on dataset titles and mentions it did not yet see. A false negative implied that BERT did not find a match, so there is no prediction, a full true positive implies that the prediction is completely correct, full wrong implies a completely incorrect prediction and good and wrong implies that at least one of the the predictions is correct and at least one is incorrect.

Results

Experiments we ran can be divided into several different categories. We ran experiments with literal matching, data augmentation, selecting the amount of capitalized words in sentences and equalizing the label occurrence in the train

set. For all of these experiments we retrained our BERT model. Through these experiments we validated what the changes were and if we saw improvement in performance.

Literal matching We started with the literal matching method, because this was in the baseline notebook. The final runs of this notebook always gave a score of **0.533**, which tells us that a certain percentage (likely about 53%) of the hidden test data has dataset labels that are also present in the train data. However, this is not a solid solution for getting a good score, since the goal of the challenge is to find datasets in the scientific papers that have not been seen yet in the training data.

Capitalized words When experimenting with this approach we introduced two variants to run the BERT model on, which were based on the amount of capitalized words in the sentence. The versions searched for either 2, 3 or 4 capitalized words per sentence. All of these versions performed less than the literal matching method. The one with 2 capitalized words performed best. The more capitalized words it filtered, the worse the performance was. The possible explanation for this was that it also filtered a lot of sentences away in which were datasets that differed from the pattern we analyzed beforehand, thus being excluded from the evaluation by the model. This causes the model to leave the prediction empty, which impacts the score negatively. The scores of these versions we submitted can be found in Table 1.

Minimal Capitalized Word count	Challenge Score
2	**0.419**
3	0.412
4	0.397

Table 1: Challenge score per minimal word count

Equalize label amount In general there were 45 unique datasets to be found in the scientific papers of the train set with an average of 346 per dataset. However, the distribution was far from equal, so we straightened this distribution and trained our BERT model again. When submitting this model without any literal matching to the challenge we got a score of **0.387**. So this approach performed slightly lower than the lowest outcome of the capitalized words approach. On the other hand, on the validation set we noticed better performance for the labels that only occur once in the validation set. As we can see in Figure 2a, with labels not equalised BERT has trouble predicting and in 83% of the cases it does not predict any label. In Figure 2b we see the results with equal labels, in which BERT never predicts a false negative. The full true positives go up from 4% to over 60%.

```
False_negative % =  0.8260869565217391
Full_true_positive % =  0.043478260869565216
Full_wrong % =  0.08695652173913043
Good_and_wrong % =  0.043478260869565216
```

(a) With labels not equalised

```
False_negative % =  0.0
Full_true_positive % =  0.6086956521739131
Full_wrong % =  0.17391304347826086
Good_and_wrong % =  0.21739130434782608
```

(b) With labels equalised

Figure 2: BERT predictions on label amount

Data Augmentation - Synonyms & Translations Finally, we applied the data augmentation methods. The BERT model seemed to have trouble predicting labels it had not yet seen before, so we thought that it could be useful to feed it with more different labels in the train set. New labels were formed based on translating back and forth and set synonyms for words in the dataset names. With this broader set of labels we retrained the BERT model. For the synonyms, in the beginning there were 45 labels and after this process 88 new labels were created. For the translation the number of labels also became 88. The results from this augmentation method performed similar to the other methods and still has a hard time finding dataset labels that it has not yet seen before. We did not submit a version on this but we did compare this method with the other methods in the Comparison paragraph.

Comparison The literal matching would fail on our validation set because it cannot match an unknown dataset. For the other methods we saw that BERT had problems with the left out dataset in every method we applied. An example of this performance on the "Equalize label amount" method can be seen in Figure 3. We see that the left out dataset is nearly completely predicted with false negatives, meaning that it could not find any datasets in the papers belonging to this label. This same behavior we saw at all of the methods that used BERT to predict when using this validation method.

	label	all_labels_count	false_negatives	full_true_positive	full_wrong	good_and_wrong
0	ADNI	374	5.0	125.0	0.0	244.0
1	Alzheimer's Disease Neuroimaging Initiative (A...	241	0.0	0.0	21.0	220.0
2	Our World in Data	212	206.0	0.0	6.0	0.0
3	Early Childhood Longitudinal Study	109	0.0	95.0	0.0	14.0
4	Baltimore Longitudinal Study of Aging	107	3.0	101.0	0.0	3.0
5	Trends in International Mathematics and Scienc...	101	1.0	95.0	0.0	5.0
6	Education Longitudinal Study	67	1.0	6.0	1.0	59.0

Figure 3: BERT predictions on left out dataset (Equal label frequency)

Discussion

Despite our different approaches to improve the BERT model's performance, we didn't manage to improve the score past the literal matching approach. Through the results we got from the different methods applied, we also dared to cautiously conclude that the BERT model as implemented in the notebooks behaved as a more fancy way of literal matching, although underperforming, since it had a lot of trouble predicting datasets that it had not yet seen before. Even though we tried to reduce overfitting on the existing labels through the data augmentation and making a more varied set of labels, it still mismatched most (if not all) of the dataset names it did not see before.

Another thing that could have made a difference in performance might be adding external datasets with dataset names not in the training set and train the BERT model on this. This would increment the known labels for the BERT model but that reinforces the idea that the BERT model would only predict things it has already seen and thus being a fancy literal matching method.

Conclusion

During this project we worked with Bidirectional Encoder Representations from Transformers (BERT) to see if we could predict datasets used in scientific papers. We did this by training the default BERT model we found in one of the Kaggle notebooks with data we altered beforehand through several different methods. Eventually, our general approach for the problem became to compare in what way the BERT model would react to these different models and we kept the BERT model itself mainly as it was. We only pre-trained it for every different method. Although we saw differences through the methods used, we did not manage to outperform the literal matching according to the competition score. This made us believe that the BERT approach still has a very hard time with datasets which it is not aware of and also makes several mistakes with known dataset references.

Author Contributions

We held general meetings once or twice a week since the start of the project. In these meetings we were all present and did our fair share of work. At the end of the project we all ran some last methods to compare them. After that Saskia and Bono started writing the report with these results and the theory. Matthijs structured the code to make it readable and ready to hand in.

Evaluation of the Process

From the start of the project we as a team met once or twice a week to discuss new finding and work on the project together. After every meeting we divided tasks on which we could work until the next meeting and discussed the progress on those tasks during these meetings. This process went quite well, because we could all do our individual parts and could ask each other questions during the meetings to realign the process. One of the things we could improve on is broadening our view a bit more from the start of the project. There were some times we followed a path towards a possible solution which in the end lead to no improvements. This caused that we eventually did not see a lot of improvement in our model. On the other side we have to state that this was our very first time working with NLP and BERT and we only had a minor introduction to this topic. This lead to us sometimes not fully understanding what was happening. We made our choices on our common sense as a team and were sometimes surprised with the results. We of course had some help with this (more on that in the Evaluation of the Supervision part), but now and then we had to learn things the hard way by reaching a dead end.

Evaluation of the Supervision

During the challenge there were two meetings with the coach and two with a teacher. These meeting were useful, as they helped us into the right direction. During the coach meetings we mostly noticed that the other team, using the same notebook, in our group was struggling with the same thing as we were. They were using the same notebook but implemented sciBERT. The other group implemented the entire notebook by themselves so it was harder to collaborate with them. Our coach Alex was helpful and gave us some good tips. At our final teacher meeting we also received some good tips on our approach. We talked about the data augmentation that we were doing but that we were also having troubles with understanding BERT completely. The teacher then told us to focus on the data augmentation and keep the BERT model simple. Where other groups may have decided to go for the best model, we tried to find the best data augmentation technique.

References

[1] Kaggle, "Coleridge initiavie - show us the data," [Online]. Available: `https://www.kaggle.com/c/coleridgeinitiative-show-us-the-data`.

[2] T. M. Phung, "Coleridge: Matching + bert ner," 2021. [Online]. Available: `https://www.kaggle.com/tungmphung/coleridge-matching-bert-ner`.

[3] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[4] X. Dai and H. Adel, "An analysis of simple data augmentation for named entity recognition," [Online]. Available: `https://www.aclweb.org/anthology/2020.coling-main.343.pdf`.

[5] S. ES, "Data augmentation in nlp: Best practices from a kaggle master," 2021. [Online]. Available: `https://neptune.ai/blog/data-augmentation-nlp`.