Named Entity Recognition Using Transfer Learning

Deep Mori
Loyola Marymount University
dmori@lion.lmu.edu

Harrison Roskopp
Loyola Marymount University
hroskopp@lion.lmu.edu

Shivani Gowda KS
Loyola Marymount University
sks@lion.lmu.edu

Abstract

In this paper we will be exploring transfer learning approaches. Specifically, we finetune BERT language model on data set from fields like biomedical, Lunar and planetary science.

1 Introduction

Biomedical and planetary science text mining is becoming increasingly important as the number of documents rapidly grows. With the progress in natural language processing (NLP), extracting valuable information from these literature has gained popularity among researchers, and deep learning has boosted the development of effective text mining models. One of the effective approaches are transfer leaning, where transfer learning in machine learning is when the knowledge learned from previous training is used to help perform a new task. Training new machine learning models from scratch can be resource-intensive, so transfer learning approaches saves both resources and time. Moreover, Transfer learning models are more generalized which means that the models are not being rigidly tied to a training data-set. Models developed in this way can be utilized in changing conditions and with different datasets.

In this paper, we investigate how the pre-trained language model BERT (Vaswani et al., 2017) behaves for biomedical and planetary science dataset.

2 Experimental setup

In this section we will be discussing the baseline model that we referenced, the dataset, and the methods.

2.1 Baseline model and the dataset

The data-sets that we used to fine-tune BERT are as following, the LPSC data-set (Wagstaff et al., 2017)[1], the JNLPBA data-set and the BC2GM (Lee et al., 2020)[2] data-set. For the LPSC data-set, the baseline models consisted of a list based NER model and the Stanford CoreNLP NER model (Wagstaff et al., 2017). The tag categories in the LPSC data-set are target, mineral, element, and others. The JNLPBA data-set contains 6 unique labels mainly protein, DNA, cell-type, cell-line, RNA, and other. Whereas the BC2GM dataset consisted of 2 tag categories that was Gene and others.

2.2 Methods

We used Google Colab notebook along with GPU backend to perform the following steps

1. Installing Huggingface's transformers library
2. Familiarizing the use of Transformers Library
3. Loading the dataset
4. Preprocessing to make it compatible with BERT as a parent model.
5. Finetune the pretrained model with suitable hyper parameters
6. Make predictions
7. Compare the results(F1 score)

3 Data preprocessing

The LPSC data-set was comprised of 62 .txt (paragraph) and 62 .ann (annotation) training and 52 .txt and 52 .ann test files that were in Brat annotation format, which was then converted to IOB format. Since each file in the LPSC data-set contained more than 10,000 words, the number of words in each sentence was limited to 50– which resulted in 2,038 traning sentences in total.

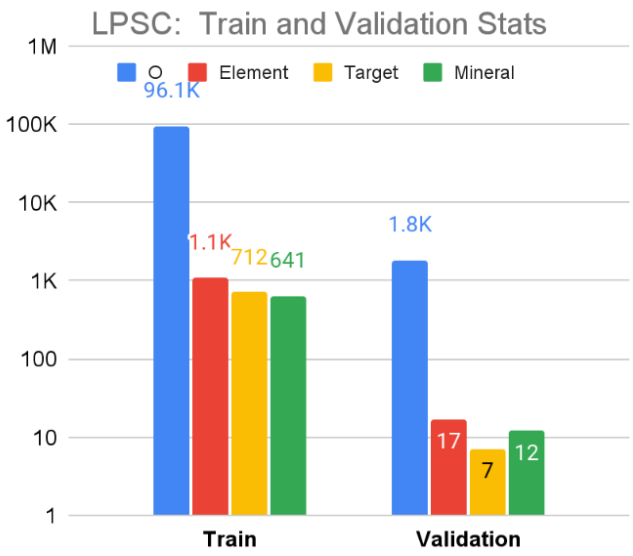

Figure 1: LPSC tags distribution of training and validation datset

[1]https://github.com/wkiri/MTE/tree/master/corpus-LPSC
[2]https://github.com/cambridgeltl/MTL-Bioinformatics-2016/tree/master/data

The BC2GM and JNLPBA data-set, were in IOB format and consisted of biomedical data. The JNLPBA training set comprised of 16,648 sentences and the validation set comprised of 3,773 sentences.

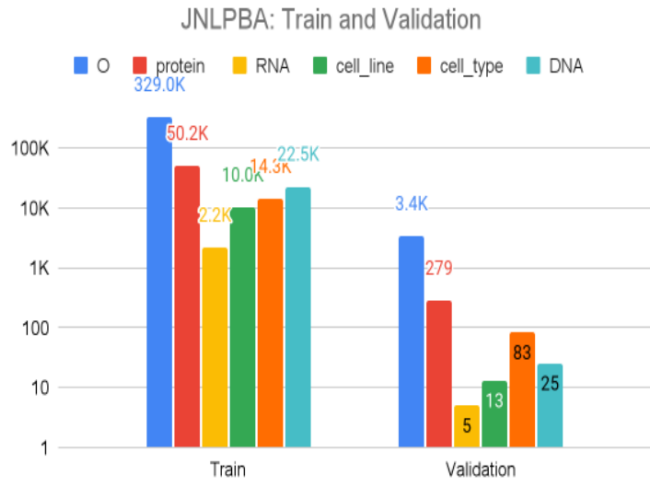

Figure 2: JNLPBA tags distribution of training and validation datset

The BC2GM training dataset is comprised of 13687 and the validation set comprised of 5,058 sentences.

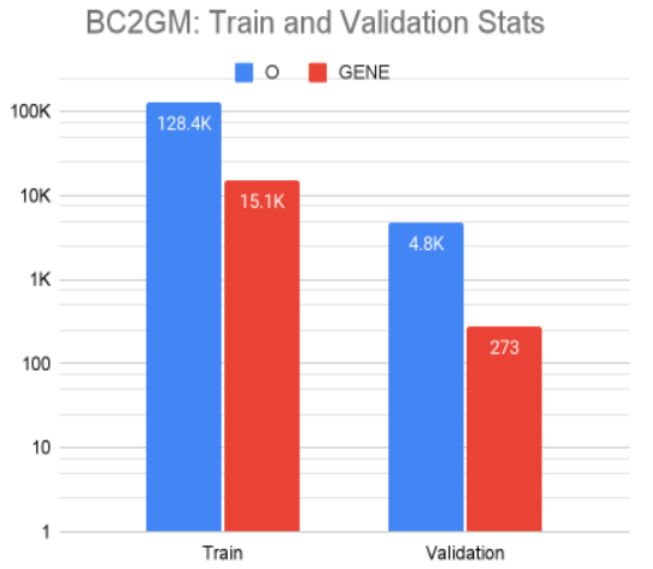

Figure 3: BC2GM tags distribution of training and validation dataset

4 Model

BERT model BertForTokenClassification is instantiated from Huggingface's transformers library's (Wolf et al., 2020). For all 3 datasets, we used similar hyperparameters except for learning rate and epochs. Model details:

1. bert-base-uncased model
2. Train and test data - 80:20 split
3. Optimizer - Adam optimizer
4. Cross entropy loss
5. Batch size - 4
6. Drop-out probability - 20

Datset	Epochs	Learning Rate
LPSC	10	2e-05
JNLPBA	9	1e-05
BC2GM	5	2e-03

Table 1: Number of epochs and learning rate for respective dataset

5 Results

The F1-score for each tag was the evaluation metric that was utilized since accuracy doesn't provide any useful insights into the performance of a model when dealing with imbalanced data. Moreover, the F1-score relies on both the precision and recall, which means that both the number of prediction errors and the type of errors are accounted for by a single metric. For LPSC dataset we trained the model on LPSC_15 and tested it on LPSC_16.

Tags	List base	Core-NLP	BERT
Mineral	0.93	0.91	0.78
Element	0.90	0.91	0.73
Target	0.63	0.45	0.46

Table 2: F1 score of LPSC dataset

While compared to that of the list based NER and Core-NLP NER, BERT results were comparatively lower.

The F1-score for tags in BC2GM dataset is around 0.81. Where as the F1-score for tags in JNLPBA dataset is as shown below. The model seems to do well except

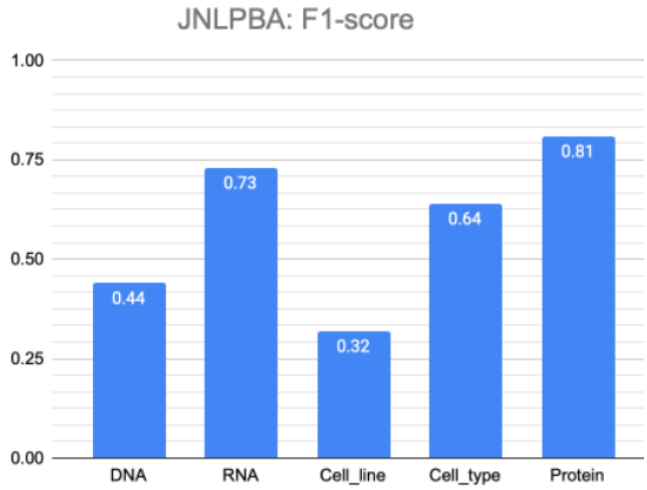

Figure 4: JNLPBA F1-score on validation dataset

for the cell_line tag, which is comparatively lower than others.

5.1 Discussion

For LPSC dataset, the F1-score for all the 3 tags are comparatively lesser than the base model, for JNLPBA dataset, one of the tag (cell_line) did not perform well, we believe this is because of the imbalance distribution of the tags and since there were 6 tags in JNLPBA dataset, it was easy to get confused. For BC2GM dataset, it performed well, which had only 2 types of tags. As per the results above, lesser the number of tag, lesser will be the confusion and better will be the F1 score.

6 Conslusion

When we combined LPSC_15 and LPSc_16 we got comparitively better results than using LPSC_15 for training and LPSC_16 for validation, even JNLPBA dataset had only around 16,000 sentences. Since these dataset are not very general BERT is not able to classify it well, may be increasing the amount of dataset with proper distribution of the tags could boost the F1-score of all the tags.

7 Acknowledgements

We are grateful to Prof. Mandy for Natural Language Processing classes, feedbacks and opportunity to learn through projects and assignments. We Thank Dr. Thamme Gowda for bar2ner.py code, Niels Rogge for tutorials on NER with BERT, Google for providing free GPU's, HuggingFace for free API's with documentation.

References

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

L Wagstaff, Raymond Francis, Thamme Gowda, You Lu, Ellen Riloff, and Karanjeet Singh. 2017. Mars target encyclopedia: Information extraction for planetary science.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online, October. Association for Computational Linguistics.