# NLP Project Update

**Mehdi Drissi** and **Vivaswat Ojha** and **Pedro Sandoval**
Harvey Mudd College, CA
{mdrissi, vmojha, psandovalsegura}@hmc.edu

## 1 Project Description

Our goal for our final project is to use transfer learning to fine-tune a model for our document classification task. Recently, a language model called BERT was released that was intended to learn useful representations for a variety of tasks. As that model was able to achieve state of the art performance in multiple NLP tasks after being fine-tuned, we hope that it will also perform well on the hyper partisan sentiment classification as well. We also plan on using semi-supervised learning. Only a small amount of the training data was hand labeled and we only plan on using those labels. The rest of the training data we will use to pre-train the model in an unsupervised manner.

## 2 Literature Review

We have familiarized ourselves with the following publications to both familiarize ourselves with the BERT model and begin thinking about how we will tweak the model for our classification task.

The papers we read are:

- Devlin, Jacob, et al. "Bert: Pre-training of deep bidirectional transformers for language understanding." arXiv preprint arXiv:1810.04805 (2018).

  This is the primary paper we are relying on this project. Its main objective was to develop the language representation model BERT (Bidirectional Encoder Represntations from Transformers) that can be easily fine-tuned for many different tasks. We plan on using this fine-tuning feature to create an effective model for our classification task. We will be using the model in a similar way to the paper's examples but intend to adapt it as necessary to suit our project. The methodology of the paper seems quite sound and it seems logically consistent. Their results show that BERT outperforms the state of the art for a number of NLP tasks that they examine. We believe we should be able to adapt BERT to perform similarly well in classifying hyper-partisan articles as well.

- Vaswani, Ashish, et al. "Attention is all you need." Advances in Neural Information Processing Systems. 2017.

  In this paper, the authors develop a neural network architecture called the Transformer based on attention mechanisms. It is important to our topic because it is one of the papers that contributed to the development of the BERT model we are using. Our work is not very similar to the authors of this paper - while they were also using their model on language tasks, the main importance of this paper for our work is its contribution to BERT. The methodology of the paper makes sense on the whole, especially given it is successfully used for the BERT paper as well. The paper's main results are performing well on certain machine learning tasks with this model. We will make use of its contributions to BERT for our project.

- Dai, Andrew M., and Quoc V. Le. "Semi-supervised sequence learning." Advances in neural information processing systems. 2015.

  This paper examines to approaches for pre-training a neural model on a large amount of unlabeled data to improve performance on a smaller amount of labeled data. One approach is to pretrain the model to be a language model while the other is to pretrain it to be a sequence autoencoder. BERT, the model we are working with, was also pre-

trained as a language model. We intend to further pretrain it using the large 600k training dataset. We do not plan to use the 600k with their labels due to the low quality of the labels. This should be helpful as the well labeled data is only around 600. The paper found both pretraining methods to work and they found the sequence autoencoder to perform slightly better. One major distinction in the BERT language model vs the langauge models they examined is that BERT was a bidirectional language model which led to the need of masking a few tokens during training (essentially filling in blanks).

- Pan, Sinno Jialin, and Qiang Yang. "A survey on transfer learning." IEEE Transactions on knowledge and data engineering 22.10 (2010): 1345-1359.

  This paper examined transfer learning in classification, regression, and clustering. The main motivation of reading this paper was getting a broader sense as to the usage of transfer learning to make sure we were not missing any major well known idea useful in other tasks. The paper covered a good deal of ground, but we did not find much useful information not already covered in the other papers.

- Sennrich, Rico, Barry Haddow, and Alexandra Birch. "Neural machine translation of rare words with subword units." arXiv preprint arXiv:1508.07909 (2015).

  This paper examined a tokenization strategy. You begin by starting off with your dataset as characters. Then you find the most common bigram and combine them into a new word piece, with the caveat that you do not merge pieces with spaces. You repeat this process until you get to a desired vocabulary size. The benefit of this method is common words will be a part of the vocabulary. Uncommon words will be broken up into more meaningful pieces like suffixes and prefixes. We found this technique well-motivated and are currently using it as our tokenization strategy. It is necessary for us to use it as BERT was pretrained with this tokenization strategy.

## 3 Current Methodologies

The main method we are operating with right now is adjusting the BERT model for the data available and evaluating its performance.

We note that, in our work so far, we are focusing primarily on the smaller data set of 645 hand-labeled articles, both for training and for validation. Our approach has been to take the first 80% of this data set for our training set and the last 20% for the validation set. We believe that, given the test set is also expected to be hand-labeled, the 645 articles are much more representative of the final test set our model will be tested on.

Due to an intrinsic limitation of the BERT model, we are unable to consider sequences of longer than 512 word pieces for classification problems, which is significantly shorter than the longest article from the training set, which is around 6,500 words. This means that we have been operating with truncated versions of the articles where we only consider the first 100, 250 or 500 word pieces.

Note that when we refer to word pieces, we are referring to the byte-pair encoding that BERT relies on for tokenization. These can be actual words but less common words may be split into pieces. The details of this process are discussed in "Neural Machine Translation of Rare Words with Subword Units" by Sennrich et al.

The main method that we intend to work with is training the model further through unsupervised training on the larger data set of 600,000 articles. We hope that this will be able to increase our accuracy on the hand-labeled validation set further. We may also explore a technique for using the entire article instead of 500 word pieces by aggregating the model's predictions for each 500 word pieces. The main weakness of this aggregation method is computational time and memory costs, but it should be doable as long as we restrict to the 80% of the 645 hand-labeled articles.

## 4 Updated results

In Lab 7, we built a baseline multinomial naive bayes model which had a cross validation accuracy of nearly 80 percent. In an attempt to improve this model, we built an extension of in Lab 8. Our extension included the generation of summary outlines for our training and test articles. Our extension did not perform as well as the baseline from Lab 7.

| Model | Cross-validation Acc |
|---|---|
| Lab 7 MNB | 78 |
| Lab 8 MNB (Outlines) | 71 |

Table 1: Lab 7 and Lab 8 results

For our current model, the results we have are based on the validation set we constructed using the last 20% of the hand-labeled articles. The main results we can report currently are the accuracies of the different versions of the model on our validation set with sequence lengths of 100, 250 or 500. These accuracies are not directly comparable to the prior accuracies given they measure on a different piece of validation data.

| Max Seq Length | Validation Acc |
|---|---|
| 500 | 79.1 |
| 100 | 76.7 |
| 250 | 75.9 |

Table 2: Accuracy by Sequence Length