News Category Classification

Benjamin Barnett	*Yaniv Bronshtein*
• Found and balanced the data • Built pipeline for base models • Researched fine-tuning methods to improve LSTM & DistilBERT output • Applied fine-tuning with Yaniv	• Preprocessed summary text • Built pipeline for Unidirectional Tensorflow LSTM model • Implemented multiclass DistilBERT • Applied fine-tuning with Ben

Introduction

With ever larger quantities of data and information, classification is increasingly crucial in today's world. In particular to news, there are a lot of sources that generate immense bodies of articles on a daily basis. However, articles are sometimes mislabeled or do not have a label. Given this, we wanted to find a way to categorize news articles using only their titles and short descriptions (rather than the entire writing), as there may otherwise be too much information to handle at a time.

To find models that can properly label news articles, we applied deep learning to predict article topics. Taking a balanced data set on six distinct topics for a total of 6000 articles, we cleaned and split the data into 80% training and 20% testing and applied the TF-IDF scoring approach for different base models, including the Naive Bayes classifier and the multilayer hidden neural network (three different MLP networks 1-3 layers were applied, with 100 neurons for each layer). Thereafter, we implemented a Tensorflow model with an embedding layer and a unidirectional LSTM layer, along with DistilBERT (using a 80 20 validation split). After fine-tuning these two models, we obtained decent performance with the LSTM model, but got even better performance with DistilBERT at approximately 90% accuracy. Beating the classification accuracy of the base models, we are confident that DistilBERT can help categorize future unlabeled articles.

Related Work (http://aclanthology.org/2020.aespen-1.4.pdf)

News category classification is a topic of interest in academic research. One such study, titled *Analyzing ELMo and DistilBERT on Socio-political News Classification*, compares ELMo and DistilBERT performance on two separate tasks. The first task trained models on local newspapers of India and evaluated performance on local Chinese newspapers (both written in English), while the other task classified sentence-level Rotten Tomatoes movies and customer reviews as positive or negative. They apply the term “null context” to refer to the India news and movie review data, and the term “cross context” to refer to the China news and customer review data. By fine-tuning the two models, the authors evaluate model performance and discover that the DistilBERT tends to generalize better than ELMo in the cross context setting. However, the authors do not find a significant difference between the models in the null context. Despite a 30% smaller embedding size and 83% faster training time than ELMo, DistilBERT still performs better than ELMo in the cross context setting. Given this result, DistilBERT appears to be very promising for tasks related to binary classification.

Data Collection

We collected around 200000 articles posted from 2012 to 2018 by HuffPost. The articles were scraped from this site: http://www.kaggle.com/rmisra/news-category-dataset. We loaded the data into a Jupyter Notebook, and found that it was very unbalanced. As an example, the politics category had over 30,000 articles, while the sports category had fewer than 5000 articles. We also found some of the categories to overlap, such as the “Arts” and “Arts and Culture” categories.

To solve these two issues, we decided to select the first 1000 observations of six distinct but popular categories, for a total of 6000 observations. These 6 categories are politics, entertainment, travel, business, sports, and religion. We then combined the article titles with their short descriptions into a single text column for classification, and removed any stopwords, punctuation, and garbage characters after lowercasing. For the non-baseline

models, we tokenized the training data (replacing unknown words with UNK tokens) and left padded the encoded sequences from the training and testing summaries. Please see below for the original and balanced data, revealed on the left and right respectively.

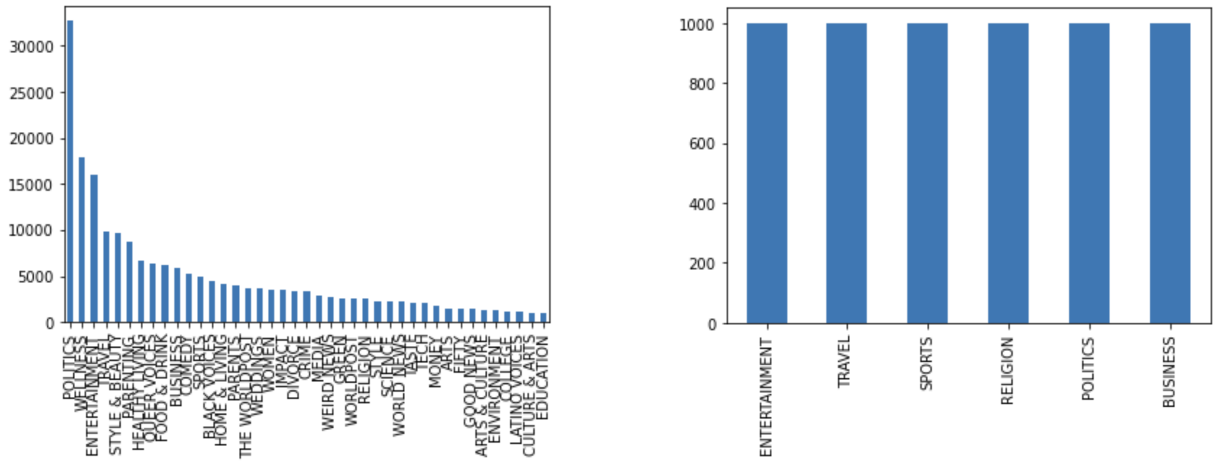

Figure 1. Data before and after balancing. A total of 6000 articles were selected for our study, using 1000 articles from entertainment, travel, sports, religion, politics, and business.

Experiments

For our base models, we used TF-IDF vectorization prior to model training. Though we attempted to use GloVe embeddings, we found that this led to poor results for accuracy and determined that TF-IDF should suffice as an appropriate baseline for the LSTM and DistilBERT models. As evident in **Figure 2**, we see that the 4 baseline models, namely Multinomial Naive Bayes and three variations of the MLP classifier (1-3 layers with 100 neurons each), all have similar performance, as the accuracy score is slightly above 0.8 for each. The confusion matrices also share close raw frequencies. Correctly classified business articles ranged from 139 to 144, and correctly classified entertainment articles ranged from 169 to 174, for instance. Holistically, while the MLP classifier with 2 hidden layers had the best performance of the base models, we feel that any can appropriately be used as a baseline. See below for their respective confusion matrices.

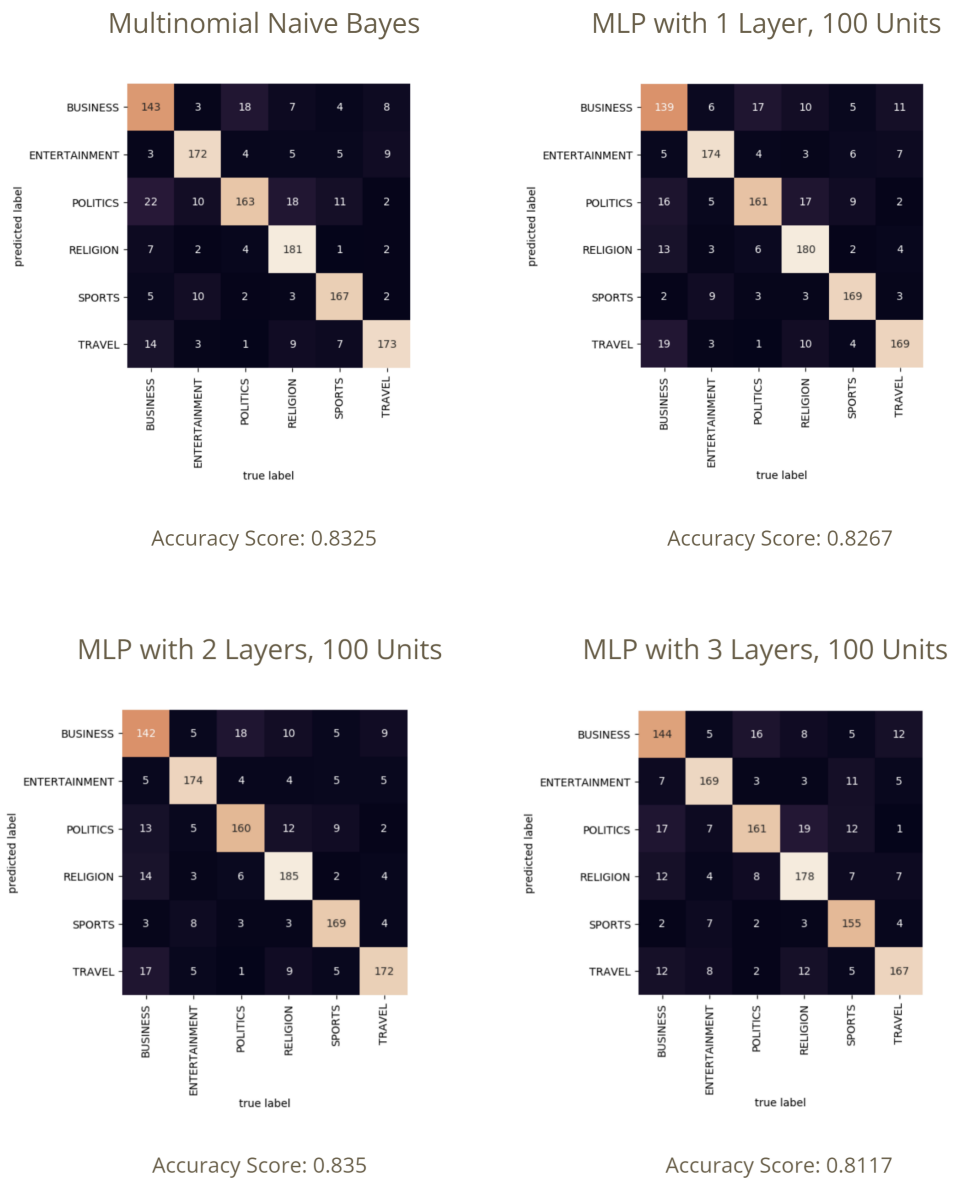

Figure 2. Confusion matrices and accuracy scores of the Multinomial Naive Bayes and three multilayer perceptron neural networks. They have very similar performance, with over 0.8 accuracy.

After fitting our base models, we proceeded with a sequential model with an embedding layer, unidirectional LSTM layer (100 nodes with 0.2 dropout), and a dense layer with 6 nodes and softmax activation. Using the adam optimizer, a batch size of 64, 10 epochs, and 20% validation split on 80% training data with an embedding size of 100 (along with categorical cross-entropy for our loss function), we concluded from its performance that more could be done to improve the model. See below for the structure of the first model.

```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding (Embedding)       (None, 81, 100)           5000000

 lstm (LSTM)                 (None, 100)               80400

 dense (Dense)               (None, 6)                 606

=================================================================
Total params: 5,081,006
Trainable params: 5,081,006
Non-trainable params: 0
_________________________________________________________________
```

Figure 3. Summary object of Tensorflow sequential model with 1 embedding layer, 1 LSTM layer with 100 nodes, and 1 dense output layer with 6 nodes.

As shown in **Figure 4a** below, the model performs well on the training set, since the loss quickly approaches zero with near perfect accuracy. However, the reduction in loss was not that stable on the testing set, and even started to increase after a few epochs. Thus, we needed to find a way to reduce overfitting for this model.

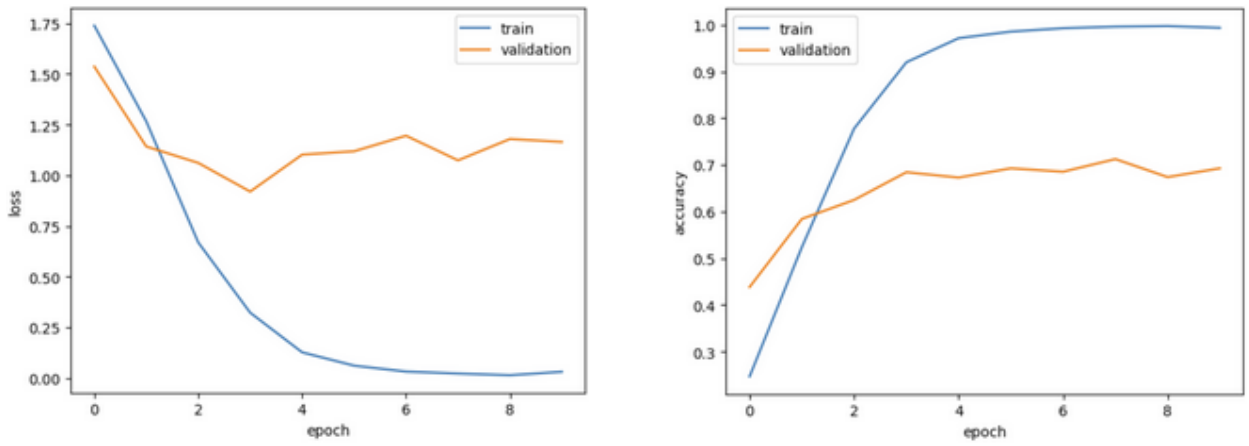

Figure 4a. Loss and accuracy curves for the LSTM model with no form of regularization.

After experimenting with different models, we found that regularization helped stabilize loss reduction and slightly improved classification accuracy. Using L1 kernel_regularizer set to 0.01 and L2 kernel_regularizer set to 0.01 (with otherwise the same structure as the first model), we improved the validation accuracy from 0.6927 to 0.7188 seen below.

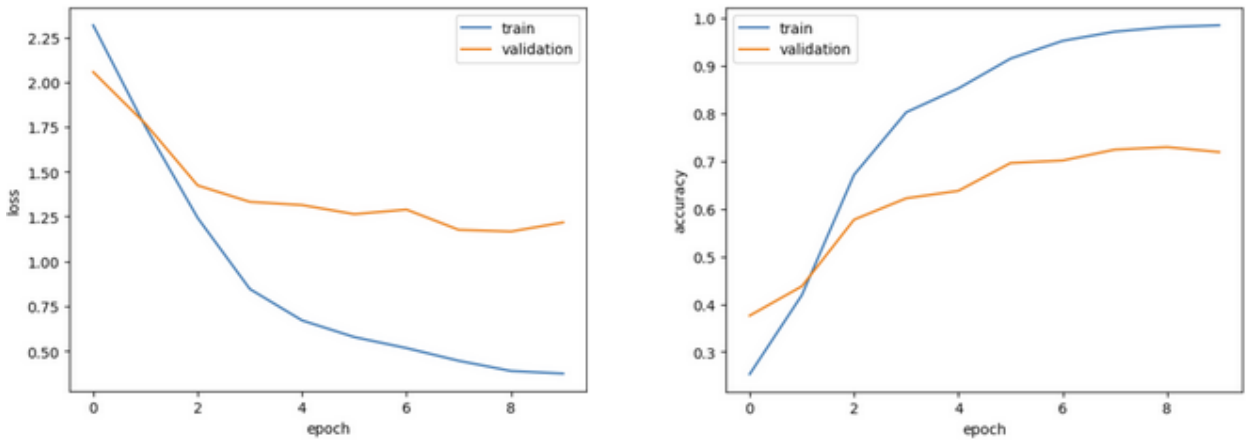

Figure 4b. Loss and accuracy curves for the LSTM model with additional regularization.

While we were able to improve the performance of the unidirectional LSTM, we wanted to further improve classification accuracy. To achieve this, we decided to implement the multiclass DistilBERT. Applying L1 and L2 regularization set to 0.01 each like before, we implemented 0.2 dropout for both the model and its attention. (We also defined a max of 128 words to tokenize and set max length padding.) However, we found the training and validation performance to be rather strange, as the training accuracy was always lower per epoch than that of validation. **Figure 5a** shows the test results below.

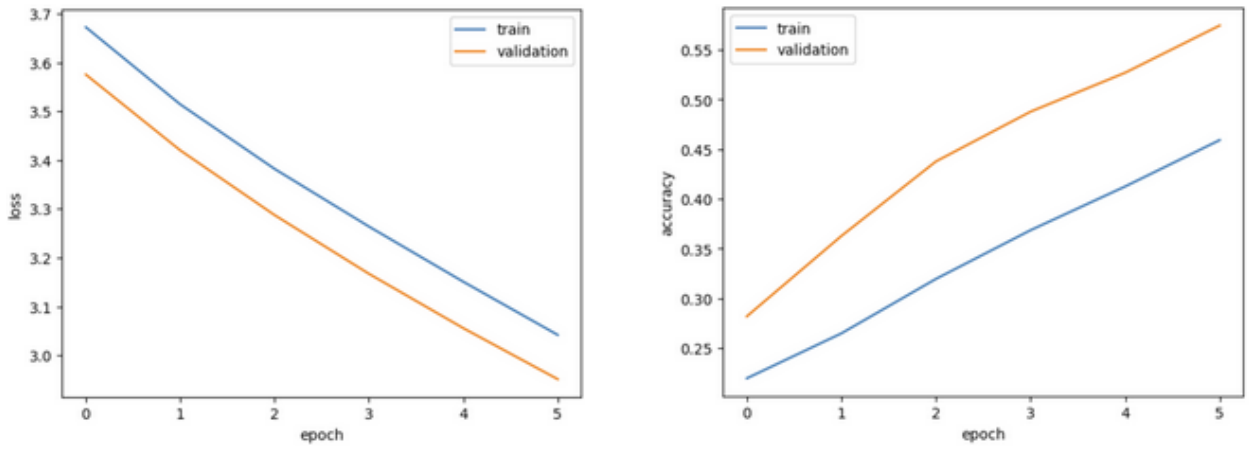

Figure 5a. Loss and accuracy curves for DistilBERT (before tuning).

After examining different parameters of the DistilBERT, we suspected that the learning rate may have contributed to the results. We changed the learning rate from the default to 5e-5, and also made all layers available for training (the former DistilBERT model had trainable layers set to false). The resulting output is below.

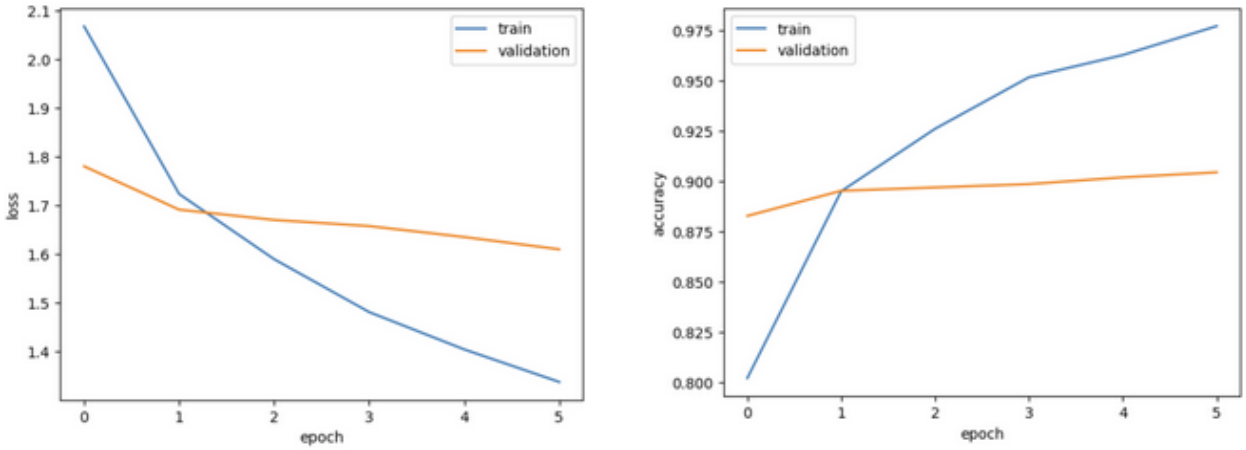

Figure 5b. Loss and accuracy curves for DistilBERT after doing tuning.

As we observe above, the classification accuracy is much better than before. At the final epoch, the validation accuracy is at 0.9042 and outperforms all of the other models (see the figure below). Furthermore, overfitting does not seem to be an issue.

Naive Bayes	Default MLP	2 Hidden MLP	3 Hidden MLP
0.8325	0.8267	0.8350	0.8117

LSTM	Regularized LSTM	Regularized BERT	Regularized Tuned BERT
0.6927	0.7188	0.5742	0.9042

Figure 6. Accuracy scores for all the models. Regularized Tuned DistilBERT clearly outperforms the rest.

Conclusions

Overall, we learned a great deal from this project. By testing the parameters of different models through trial and error, we not only discovered how to prevent overfitting through L1 and L2 regularization but also how to apply DistilBERT model for the multiclass case and improve its performance. Furthermore, we learned that GloVe embeddings may not always be appropriate to use, as TF-IDF vectorization led to much stronger accuracy for the base models. Given that we only worked with article titles and short descriptions, the GloVe embeddings may have done better had we worked with full text. Nonetheless, the Multinomial Bayes classifier (and the other 3 MLP networks) all had over 80% accuracy by simply using the TF-IDF approach.

One thing we could have done to further improve the LSTM was make it bidirectional. It would have been interesting to evaluate and compare its performance to the remaining LSTM and DistilBERT models. Though its accuracy would have probably increased, we also expect that it would have needed regularization to reduce overfitting. For the single directional LSTM model, we found that balanced L1 and L2 values helped stabilize loss reduction more than when L1 and L2 were very unbalanced.

We spent a lot of time and effort not only trying different models, but also just getting the multiclass DistilBERT model to work. While we had little issue changing the dense layer to output 6 nodes and the loss to categorical cross-entropy, we took a while to discover that we had to use maximum length padding and not dynamic padding (set to longest at default) to obtain equal embedding sizes. Once we fixed this issue, we were able to run our DistilBERT models successfully. Since these models took longer to execute than the LSTM models, we only plotted the results of the first 6 epochs. Though we implemented the DistilBERT model from class, evaluating the performance of other models like ELMo would also have been very interesting.

On a final note, this study could be further improved by taking more than six categories and one news source. Rather than artificially balancing the data, we could have applied

class weights to capture the data in full. Although we took articles from HuffPost, there are plenty of other news sources. Like the related work, looking into additional sources could prove insightful. Nevertheless, we have shown that the DistilBERT can categorize very well in the multiclass case and can help with future article labeling.