Assignment 3 – Authorship Attribution

Yiftach Savransky - 312141369

1 Introduction

Authorship Attribution NLP analysis of Donald Trump’s tweets. The tweets’ texts were analyzed, and features were extracted. Various models were evaluated, and their performances compared. The code of this assignment is attached in ex3_312141369.py and the full notebook can be found here.

2 Data description and assumptions

In this section the major assumptions related to the data are presented followed by descriptions of the preprocess steps that were used.
A small dataset of 3518 tweets collected from three accounts controlled by Trump (directly or indirectly) was used to train various supervised Machine Learning (ML) models. The dataset contains four features for each tweet. The user’s handle that published the tweet, the time of the publication, the device used and the tweet’s text.

3 Preprocess

The preprocessing stage consist of several central steps. In this section the preprocessing steps are presented.

3.1 Filtering

The first step is filtering tweets from before Trump’s presidency. The date Donald Trump took control over the ‘POTUS’ twitter account is the date of his inauguration – 20/1/2017[0]. This date is used to filter the tweets from both the ‘PressSec’ handle and the 'POTUS' handle. The tweets from these handles are discarded as they are from the previous administration. As we aim to train a precise classifier, these tweets may add unwanted noise to the data and may impair the learning process. These tweets may be "to easy" to separate from Trump's tweets, therefore both skewing the evaluation metrics and hindering the models' ability to distinguish between Trump’s tweets to his team’s tweets.

Additional samples filtering included the removal of tweets that had on remaining tokens after the preprocessing steps described below.

3.2 Labeling

The dataset does not contain an explicit author for each tweet. The labeling of the tweets’ author is accomplished using two main heuristics.

First, Trump used an android phone to tweet. Therefore, all the tweets that were published using an android device were labeled as written by him (2255 tweets). Second, working hours were considered. It's relatively safe to assume that any text-only tweets published after working hours were authored by Trump [01]. We defined working hours loosely as between 5am and 11pm. Tweets published during the working hours from any other device other than android were labeled as being written by Trump’s staff. Figure 1 shows the distribution of labels colored by the device used to tweet.

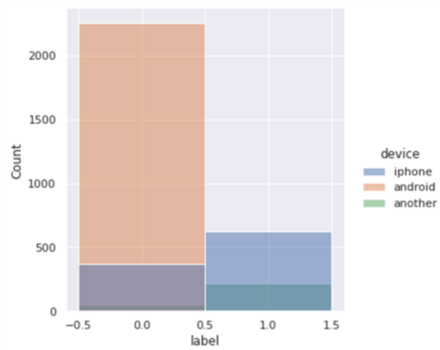

Figure 1: Labels distributions. The label 0 represents tweets labeled as being authored by Donald Trump, while the label 1 stands for tweets authored by other staff members

3.3 Text Preparation

Multiple preprocessing and cleaning techniques are used to prepare the tweets' text. They include simple string manipulations, followed by stop words removal and Part-Of-Speech (POS) Tagging used for lemmatization.

The text of each tweet is transformed to lower case and the punctuations are removed as well as any URL address. The text is tokenized using NLTK's TweetTokenizer which also removes handles from the text. Afterwards, any stop words are removed.

The next step of processing the text is lemmatization of the words' tokens. NLTK's WordNetLemmatizer is used to accomplish this task. The lemmatizer requires POS tag of each token to work correctly. Therefore, NLTK's pos_tag is used to assign POS tag to every token.

3.4 Sequencing

To facilitate the usage of Recurrent Neural Network (RNN) model, the tweets' text must be sliced into sequences of the same length. Sequence length was set as 10 (found to be preferable for performance). Tweets with less than 10 tokens were padded using a special token. Tweets with more than 10 tokens were split into consecutive slices.

4 Feature Extraction

4.1 Embeddings

Embeddings of the words in the tweets were used as the main features for the models. The embeddings, provided by FastText [2], include 2 million word vectors, 300 numbers each, trained on Common Crawl (600B tokens). Only the relevant embedding vectors were saved and used for training. Uniformly sampled vectors were generated as the embeddings for words that are out-of-vocabulary of the embedding model. The special padding token's embedding was set as a zero vector.

Each word token was encoded using the embedding model. Next, the mean embedding vector of the tweet's tokens was calculated (as the average of the tweet's tokens' embeddings). This vector is used as the input for several models instead of the full embedding sequence due to computational limitations (time and memory restrictions).

4.2 Additional features

Additional Features were extracted to be used as part of the model. The length of the tweet, the time of day of its publishment are simple features that were extracted directly from the unprocessed data. More Advanced features were extracted as well, such as TF-IDF and POS tagging of each token. However, these features were not used in the final model. This is because they did not provide any significant performance improvement.

5 Models

Various supervised ML models were trained, and their performance evaluated. Four model from the SKLearn library were used: Logistic Regression, SVM.SVC with linear kernel, SVM.SVC with Radial Basis Function (RBF) (non-linear) kernel and Ada-Boost Classifier. It is important to note that in contrast to the first three models, Ada-Boost is an ensemble model. This means that the algorithm trains multiple base classifiers and uses them to produce the final prediction. The base classifier used was Decision Tree Classifier. In addition, Two Neural Network

SKLearn Model	HP	range	HP	range	HP	range	HP	range
Logistic Regression	class_weight	None, 'balanced'	C	.001,.009,.01,.09,1,5,10,25	penalty	'l1', l2'	solver	'liblinear','saga'
SVC (Linear and Non-Linear)	class_weight	None, 'balanced'	C	.001,.009,.01,.09,1,5,10,25	decision_function_shape	'ovo', 'ovr'	gamma	'scale', 'auto'
Ada-Boost (DecisionTreeClassifier parameters)	max_features	'log2', 'sqrt'	criterion	'entropy', 'gini'	min_samples_split	2, 3, 5	min_samples_leaf	1, 5, 8
Ada-Boost	n_estimators	100, 150, 300						

Table 1: Hyper-parameters used for the grid search training of the SKLearn models. HP stands for Hyper-Parameter name.

(NN) classifiers were implemented using the PyTorch library: Feed-forward NN and LSTM.

The input to all the models, other then for the LSTM, is the embeddings mean vector of the tweet's tokens. For the LSTM model, the inputs are a sequence of length 10 of the embedding of each token.

5.1 SKLearn-based Models

To facilitate the use of models from SKLearn, a base class is implemented which acts as a uniform wrapper for them. Grid Search and 5-fold Cross Validation is used to search over the hyper-parameters range and select the best performing model. Table 1 details the hyper-parameters ranges that were defined to search over.

5.2 Neural Network Models

The Feed-Forward NN model defined has 4 fully connected layers with an input size of the embedding length (300), and an output size of 1. ReLU activations are used following each of the first three layers and a sigmoid layer used for the output. Binary Cross Entropy (BCE) loss is used, along with Stochastic Gradient Decent (SGD) optimization algorithm. In addition, learning rate decay is used to assist with the convergence of the model.

Hyper-Parameter	NN	LSTM
Input shape	(300)	(10,300)
Epochs	300	300
Batch size	16	16
Initial learning rate	0.1	0.1
Decay step	10	10
Decay rate	0.8	0.8
Layers sizes	300, 150, 75, 37, 1	(10,300), (10,300), 128, 32, 1
Hidden dimension	/	128

Table 2: Hyper-parameters used for the training of the PyTorch NN models.

The LSTM model is defined using PyTorch's implementation of the LSTM layer. Two LSTM layers are used, followed by a fully connected layer, ReLU activation, additional fully connected layer (in the size of the output – 1) and finally a sigmoid function. The input for this model is a sequence of 10 word-embeddings resulting in input size of [10,300] for each sample. As with the previous model, BCE loss, SGD and learning rate decay are used.

The models' hyperparameters are shown in Table 2.

5.3 Train-Test split

The splitting of the dataset into train and test sets was done based on the tweets' dates. The models are trained on early tweets and evaluated on more recent ones. 80/20 split was achieved by dividing the dataset using the date 27/7/2016. All the tweets before this date are part of the training set and all the tweets after are part of the testing set. Table 3 shows the details of the sets.

dataset	Size/Percentage		
	Total	Train set	Test set
Tweets	3499	2846/81%	653/19%
Split tweets	5073	4056/80%	1017/20%

Table 3: Sets details. Split tweets dataset is a data set with the same length sequences.

6 Metrics

AUC metric was chosen rather than accuracy to overcome the fact that the data is heavily imbalanced. At first, using accuracy was considered as the evaluation metric, but even a model based on majority voting will achieve high

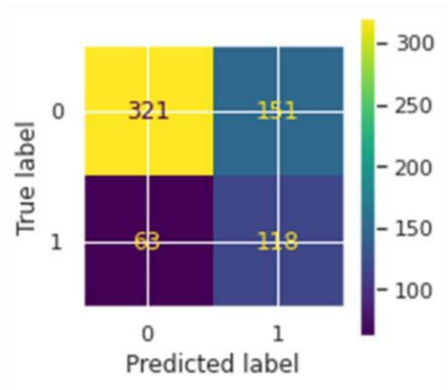

Figure 2: Confusion matrix of the best performing model.

accuracy score (always predicts 0 – Trump is the author).

The Recall metric was also considered, because a high recall score indicates that a high percentage of the Tweets not authored by Trump will be identified. Even though this metric seems to fit this problem well enough, AUC was chosen instead. AUC takes into consideration TPR and FPR while Recall (or sensitivity) regards only TPR. This is the reason it was chosen as the comparison metric.

7 Performance Results

The models presented were trained on the train set using the embeddings vectors extracted. The models' performances were then evaluated on the test set samples. A detailed performance analysis was conducted for each model. The performances of the models are presented in Table 4.

7.1 Best performing

The best performing model was selected using the AUC score – **Linear SVC**. The confusion matrix of this model can be seen in Figure 2. The hyper-parameters of the best performing model

model/ metric	AUC	Accuracy	Precision	Recall
Logistic Regression	0.647	0.65	0.62	0.65
SVC Linear	**0.666**	**0.67**	**0.64**	**0.67**
SVC Non-Linear	0.661	0.69	0.64	0.66
Ada-Boost	0.57	0.72	0.63	0.57
NN	0.62	0.72	0.64	0.62
LSTM	0.58	0.73	0.6	0.58

Table 4: The performance of the different models. SVC achieved the highest AUC score.

are: {'C': 5, 'class_weight':balanced', 'decision_function_shape': 'ovo', 'gamma': 'scale'.

8 Discussion

An interesting outcome of the evaluation is that the NN models did not perform as well as the SKLearn models. This is due to a several factors, in my opinion, the most significant is time and computational limitation. Grid Search over hyper-parameters of the NN models which included k-fold Cross Validation was implemented but not used due to the long running times. This prevented the fine tuning of the hyper parameters. Therefore, a significant optimization process is required to further improve the performances of these models.

8.1 Data insights

As part of the analysis several interesting insights can be drawn from the data. First, it is very clear from Figure 3 that the number of tweets published during the presidential campaign is reducing, reaching a minimum around the election date 8/11/2016.

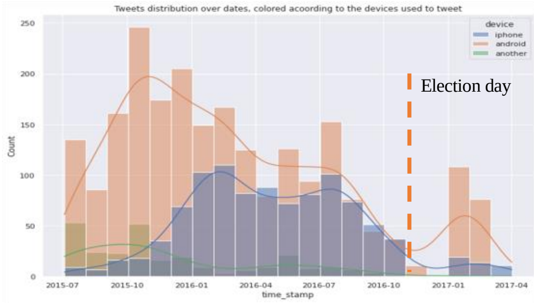

Figure 3: Tweets distribution over dates, colored according to the devices used to tweet. The election day is marked.

Additional insight regards the time of day of the publication of the tweets. Tweet labeled as Trump's are more likely to be published at the afternoon or at night, while other's tweets are published mostly in the evening or in the morning. Figure 4 shows these patterns. It should be noted that this pattern may change based on the data labeling process.

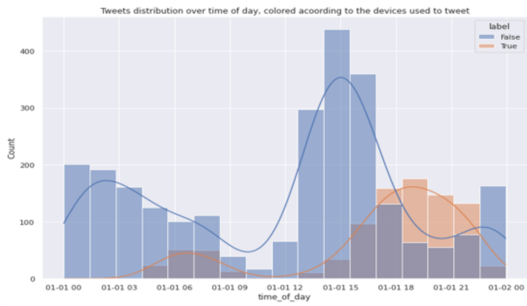

Figure 4: Tweets publications distribution over the time of day, colored according to the label.

References

Collins, T. (2017, January 20). First Trump administration tweets come from @realdonaldtrump. CNET. Retrieved December 21, 2021, from https://www.cnet.com/news/president-obama-twitter-donald-trump-hillary-clinton-twitter-inauguration/#

Feinberg, A. (2017, October 6). How to tell when someone else tweets from @realdonaldtrump.

Wired. Retrieved December 21, 2021, from https://www.wired.com/story/tell-when-someone-else-tweets-from-realdonaldtrump/

English word vectors · fasttext. fastText. (n.d.). Retrieved December 21, 2021, from https://fasttext.cc/docs/en/english-vectors.html