Natural Language Processing – Assignment 3

In this report we will review our work of classifying between Trump's or not Trump's tweets from 3 different official tweeter accounts.

Data Exploration

The data consists of 3 different users who wrote the tweets: "POTUS", "PressSec" and "realDonaldTrump". There are 9 different devices that the tweets were tweeted from. We can see the distribution of the number of tweets in the following table:

User	Device	Number of tweets
POTUS	iphone	1
PressSec	Twitter Web Client	2
	TweetDeck	8
	iphone	2
realDonaldTrump	Instagram	4
	Twitter Web Client	229
	Twitter for iPad	6
	Facebook	1
	Twitter for BlackBerry	12
	Periscope.TV	3
	android	2005
	iphone	921

The tweets in the dataset were created between: 29/4/2015 – 1/4/2017.

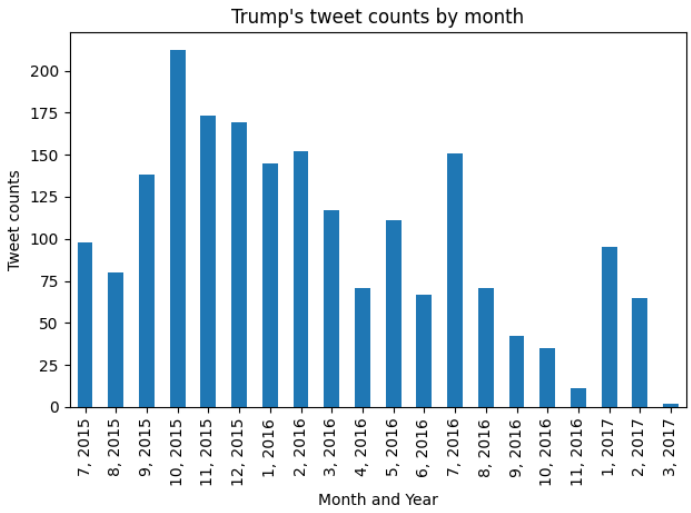
It's known that Trump had an android device until April 2017, then all tweets from an android device are assumed to be Trump's. We assume that all tweets from an iPhone device are not Trump's since he doesn't have this type of a mobile device. The rest of the devices used are mainly from other mobile devices, that Trump doesn't have, or from a web client that was probably used from a desktop – potentially by the office people that come to work at Trump's office.

We then generated the labels as follows: all android tweets are Trump's (labeled with 0) and all the rest are not Trump's (labeled with 1).

The created label distribution of tweets is: 2005 tweets labeled with 0 and 1200 tweets labeled with 1.

Was Trump restricted from his account during the campaign?

We can see in the following plot Trump's number of tweets by month:



We can see that starting on august 2016 until the end of 2016, Trump has decreased his number of tweets, and in December 2016 he didn't even tweet one time. Trump was elected in November 8th, 2016, so we can assume that he was actually kept away from his Twitter account – even a little later after he was elected.

Feature Extraction:

There are several types of features we extracted to classify these tweets with:

1. 1-grams / 2-grams / 3-grams.
2. Tf-idf.
3. Word embeddings -GloVe pre-trained embedding.
4. Meta data on the tweets – year, month, day, hour, number of words, number of chars, number of capital letters, number of hashtags, number of mentions, number of retweets, number of stopwords.

When showing the results of each model we will specify the type of features used.

Preprocessing

For preprocessing, we do various steps as follows:

1. We remove tweets that are empty after the preprocessing.

2. We removed retweets (the "rt" in the beginning of the text), mentions and hashtags
3. We could see that there are many urls shared on the tweets. We delete them so they wouldn't add noise to the words tweeted.
4. We remove punctuation.
5. We remove stop words.
6. We stem the words.
7. We lower the letters.

Data split – in the train we received: 80% Train (i.e. our train set), 20% Test (i.e. our test set)

Results

We created a train set and test set for training the models out of the given train set. All results are shown on our test set. Best model result in each model is in bold.

Logistic Regression:

Trained with 5 cross validation on our train set. We also made a grid search with the C parameter. We present the best models from the grid search on the different features generated.

Metadata features, C=1.0	1,2,3 ngrams, C=1	Tf-idf, C=10
0.729	0.744	**0.778**

SVC Model:

Trained with 5 cross validation on our train set. We trained a linear and rbf kernels. We also made a grid search with the C parameter. We present the best models from the grid search on the different features generated.

Tf-idf, C=1.0, kernel=linear	Tf-idf, C=100.0, kernel=rbf	1-gram, C=0.1, kernel=linear	1-gram, C=1.0, kernel=rbf	2-gram, C=1.0, kernel=linear	2-gram, C=10.0, kernel=rbf
0.735	0.735	0.72	**0.741**	0.707	0.647

Random Forest Model:

Trained with 5 cross validation on our train set. We also made a grid search with the max_depth parameter. We trained it on our metadata features and got the best model with max_depth=3 and accuracy of **0.768**.

Neural Network:

Trained with 80% of our train set and 20% of our train set were a validation set.

Architecture:

1. GloVe embedding of vector size of 100 (every tweet is turned into a vector of size 100) – output in size 100.
2. Linear layer - size 50.
3. ReLu layer – size 50.
4. Linear layer – size 1.

Evaluated on our test set and received accuracy of **0.619**.

LSTM:

We checked different numbers (5/10/15/20) of first words to take from every tweet. We received the best results with 10 first words taken from each tweet.

Architecture:

1. GloVe embedding of vector size of 100 (every word in the tweet is turned into a vector of size 100).
2. 1 LSTM unit, hidden layer of size 128.
3. Dropout of 0.5.
4. Fully connected layer of size 1.
5. Sigmoid layer.

Evaluated on our test set and received accuracy of **0.74**.

<u>Best Model</u>

The best model we received was logistic regression that was trained on the Tf-idf features and with a hyper parameter of C=10. It received accuracy of: **0.778**.

We have witnessed that the neural network received the worst results. It may have happened because we took an embedding of each word in the text and made an element-wise average on all of the vectors. Different types of embeddings of text to vector maybe could have worked better.