Natural Language Programming

END SEM Project

FAKE NEWS / RUMOR DETECTION

Introduction

We have created five models. For each model, we have used a different regressor and a different set of features.
To confirm our results and findings, we have trained on two separate datasets.

Pre Processing

Since we have worked with different models, we have used many preprocessing for each model.

LSTM (RNN): We have first removed all the noise in data (punctuations, numbers, unreadable characters, extra spaces, etc.). We have also removed stopwords. We have then encoded the data using One Hot encoding and trained.

CNN (DL): We have first removed all the noise in data (punctuations, numbers, unreadable characters, extra spaces, etc.). We have also removed stopwords. Then we have taken the unigram and bigram features for the sentences using CountVectorizer and trained the model on that.

MLP (ANN): We have first removed all the noise in data (punctuations, numbers, unreadable characters, extra spaces, etc.). We have also removed stopwords. Then we have taken the unigram and bigram features for the sentences using CountVectorizer. We have also considered many sentiments and lexicon analysis features for this data like BingLiu, hashtags, etc.

Lasso (Linear regressor): We have first removed all the noise in data (punctuations, numbers, unreadable characters, extra spaces, etc.). We have also removed stopwords. Then we have taken the unigram and bigram features for the sentences using

CountVectorizer. We have also considered many sentiments and lexicon analysis features for this data like BingLiu, hashtags, etc.

Decision Tree (Tree): We have removed all the noise in data (punctuations, numbers, unreadable characters, extra spaces, etc.). We have also removed stopwords. Then we have taken the unigram and bigram features for the sentences using CountVectorizer. We have also considered many sentiments and lexicon analysis features for this data like BingLiu, hashtags, etc.

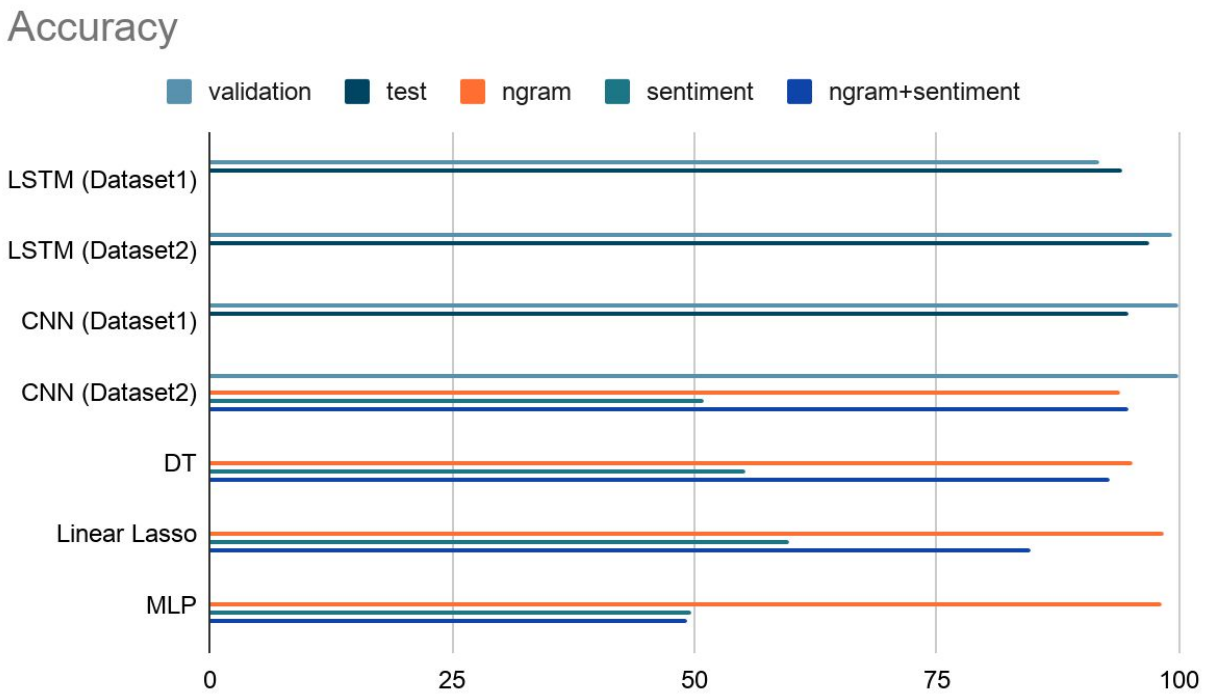
Methodology

We have created these different models knowingly for different situations. Some provide higher accuracy than others at the cost of time. Some are faster than others but not as accurate.

For some models, we have considered only the news title (which surprisingly gave excellent results on the used datasets). For a few models, we have used the actual text in the news. This was considerably slower and did not provide a very significant increase in accuracy (although the model becomes much more generic and can handle many different types of news rather than just the ones given in the dataset).

We also used an additional Whatsapp Fake forwards dataset, which was gigantic. It was used to cross-validate the results and not for training.

Results and Conclusions



From the results, we can conclude that LSTM was consistently giving the best accuracy on both the datasets.

CNN was also giving good results, albeit it took a long time to get the results

Ngram features were the best to classify news as fake or real. Sentiment and lexicon features alone were not able to determine the results. Moreover, they take a lot of time to be computed as well (especially VADER Sentiments)

Together Ngram and sentiment features gave the best results, which were consistent for all datasets.

Cross-validation on the WhatsApp forwards dataset gave slightly worse results, which are expected from a dataset containing a very different type of news.

(Additional results and confusion matrix in the python notebooks.)

Speed: Decision trees were consistently the fastest classifiers and also had good accuracy

SVM was very slow but gave slightly better results than the Linear Regressors.

LSTM took around 10 minutes to train but gave the best accuracy. (Overall best choice)

CNN was a close second but was slightly more accurate.

MLP gives good accuracy in the Countvector method within a good time.

Datasets

1. https://www.kaggle.com/c/fake-news/data
2. https://www.kaggle.com/clmentbisaillon/fake-and-real-news-dataset
3. https://github.com/sahitpj/India-WhatsAppFakeNews-Dataset

Both the datasets very wholly unbiased.

Dataset1 had a few missing values, which were ignored.

Both datasets obtained were from major US news channels and news publishers and contained a lot of general information and Political information. These will not be very suitable for news from other countries (mainly due to vocabulary issues).