Stance Detection in Online Media

Prakhar Kumar MT21066 **Amit Kumar Singh** MT21008 **Anshu Kumar** MT20050

1 Introduction

The term "Fake News" is used to represent false, misleading or fabricated information. Fake news always existed in society but with the advent of internet and social media, it is too frequent and abundantly present around us. It is spread via various channels such as television, print, social media etc. The whole purpose of spreading such news is to defame someone or mislead the readers or to gain popularity. Today, researchers have a powerful tool like stance detection to detect fake news. Using stance detection, we can find find out the relationship between two text pieces. In this study, we are focused on stance detection, given a news article body and a headline.

2 Related Work

In paper [1], word2vec embeddings of the headline and articleBody features are passed to Bi-LSTM(Bi-directional Long Short Term Memory layer. The output from two Bi-LSTM blocks are concatenated and are passed though an autoencoder. The autoencoder encodes the higher dimensional feature vector to lower-dimensional feature representation and thus learns a compressed version of input vector. These input vectors are used by dense neural network layer for classification. The denser layer contains 64 neurons and the output layer comprise of four neurons which predict the stance as one of the four classes.

Paper [2] proposes hybrid deep learning model involving two neural network layers i.e. CNN and LSTM. This hybrid deep learning model is trained on data models formed using feature reduction techniques. They have created 4 data models wherein first data model uses all features without preprocessing, second model uses preprocessed features without any feature reduction. Third and fourth models are developed using dimensionality reduction techniques including PCA and Chi-square. After the features are selected by any of the four models, these features are fed to CNN-LSTM hybrid model. In this model, first the embedding layer produces the vector for each word. These vectors are then passed to the CNN layer which extracts contextual features. The output of CNN layer is passed into LSTM and then fed into the connected dense layer to produce a single instance for each claim as output.

Thota et al. [3] used three different versions of deep neural network. First they preprocessed the dataset in which the stop words and punctions and stop words are removed and stemming was performed. In the first version, the researchers took the Tf-Idf vectors of the headline-article pair and their cosine similarity as input and predicted the output stance. Then they passed the Tf-Idf vector to a dense neural network. The final dense layer predicted the probability of the stances. The second version used Bag of Words with dense neural network architecture. BoW vectors are concatenated with cosine similarity measure and passed as input to the dense neural network. In the third version, Word2Vec is used to as word embedding which then fed into a dense layer. Out of all the three versions, Tf-Idf vectors combined with neural network performed best with accuracy 94.31%.

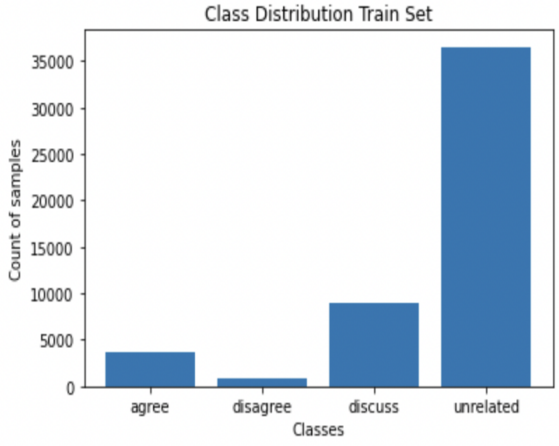
3 Dataset information

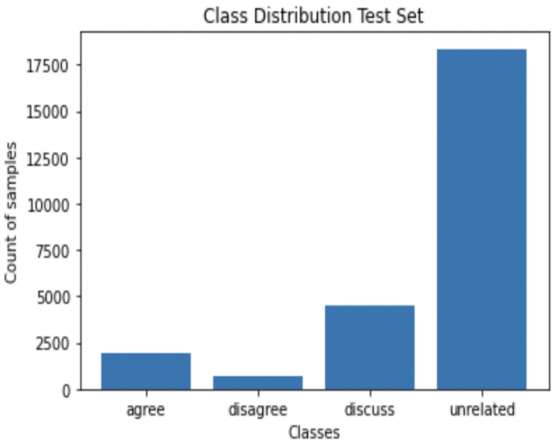
The dataset is made up of two csv files. First csv files contains Headline (text column denot-

ing headline) and Body ID column which is a mapping to the body of the Headline. Second csv file stores news text in articleBody column and Body ID column. The Body ID column is used to get the corresponding article body given a headline. Dataset target column is Stance column, there 4 unique stances which are possible (unrelated, agree, disagree and discuss). Dataset is highly imbalanced as unrelated samples are a lot more than others, samples with disagree stance are the least.

4 Data Analysis

The dataset is highly imbalanced. Both the train and test set have similar distribution.





Both of dataset contain similar types of most frequent words.

5 Methodology

- Load the data - two csv files, one contains the news headlines and other contains the corresponding body text.We join them on Body ID column to get Headline and related body text together.
- Preprocess Text - involves removing punctuations, stopword and extra spaces and newline characters which are present in textual data.
- Generate Features - we have used TFIDF and Word2Vec for one experiment and used cosine similarity, euclidean distance, manhattan distance after using TFIDF on both Headline and Body text in another experiment.
- After the data is in required format we trained multiple models like Logistic Regression, MLP etc.
- the best model is then attached with an online app from which user enters the Head-

line of news and text body of the news to checks how is body related to headline and then result is displayed.

6 Experiments

6.1 Featurization 1

Our First experiment involves featurizing both the Headlines and Body Text separately using different TFIDF Vectorizers. We also use Google News Word2Vec pretrained representation to generate word2vec features. We train models on both of these featurizations in our first experiment.

6.1.1 Logistic Regression

Logistic Regression is a Linear Model which tries to find a line(in 2D, a hyperplane in n-D) to classify between the two classes. For multiple classes, the one vs. rest strategy is used. We hyperparameter tuned the C value, and the f1 score on test data we got was 0.25 using TFIDF Features and 0.26 using Word2Vec featurization.

6.1.2 Decision Tree

Decision Tree is another model that uses a greedy heuristic to create a tree where every node judges the input sample on an attribute. Based on the comparison result, we move to another node in the tree to again compare on another attribute so on. We hyperparameter tuned the max depth parameter of the decision tree, and the max f1 score on test using TFIDF features was 0.26 and using Word2Vec features was 0.29.

6.1.3 Random Forests

Random Forest is an ensemble method, it combines multiple trees of short depths using random attributes and then takes the majority vote to get the predictions. We hyperparameter tuned the max depth of each tree in forest and pruning using complexity parameters of random forest. The f1 score on test data using TFIDF features was 0.24 and using Word2Vec was 0.23.

6.1.4 SVC

SVC is similar to Logistic Regression in the sense that it also tried to find a hyperplane that maximizes separation between the classes. SVMs are better because we can use kernel trick to get cosine distances in higher dimensions which may help in better classification of data points. F1 Score on test data was 0.25 using TFIDF features.

6.1.5 MLP

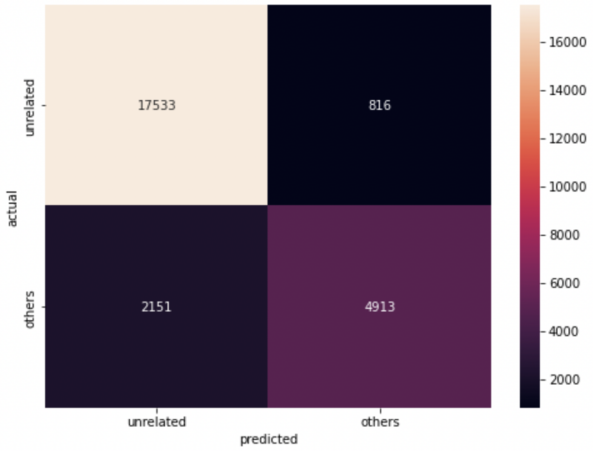
MLP is a neural network based classification technique and the most simplest one. In our MLP we used 1000 nuerons in our hidden layers and used top 3000 Features as input from TFIDF for both Headline and Body Text. The F1 score using this network on test set was 0.38.

6.2 Featurization 2

In our second experiment instead of dealing with all 4 classes at one time. We decided we will first try to classify unrelated vs others(agree, disagree, discuss). After this classification is done then any samples which have been labelled as others will under go another classification from other model which will handle classification between the three classes remaining - agree, disagree, discuss. To classify between the unrelated vs others we use cosine similarity, euclidean distance and manhattan distance between TFIDF features of both Headline and Text. This time instead of using vectorizer for both seperately we used a single vectorizer and took 1000 samples and calculated the distances as mentioned. For our second model in this pipeline we then used top 3000 features from TFIDF to classify between the 3 remaining classes.

6.2.1 Unrelated vs Others

We trained an MLP to classify between unrelated and others. The MLP had two hidden layers, one hidden layer had 50 neurons and other had 10 neurons. The F1 score for first classification was 0.77. Confusion Matrix for unrelated vs others:



CONFUSION MATRIX:

	agree	disagree	discuss	unrelated
agree	721	85	594	503
disagree	180	54	166	297
discuss	508	94	2511	1351
unrelated	203	71	542	17533

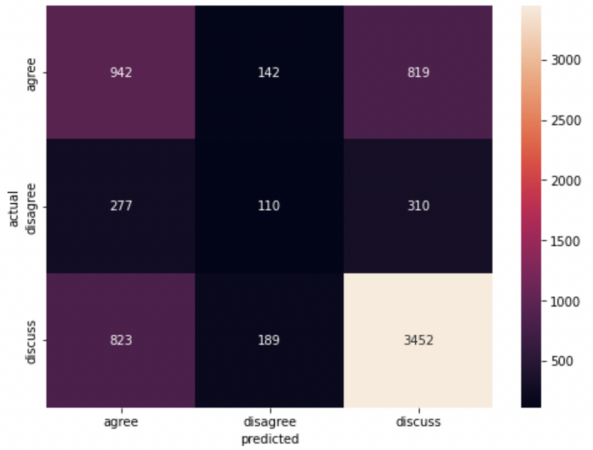
ACCURACY: 0.819

MAX - the best possible score (100% accuracy)
NULL - score as if all predicted stances were unrelated
TEST - score based on the provided predictions

MAX	NULL	TEST
11651.25	4587.25	8076.0

6.2.2 Agree vs Disagree vs Discuss

The MLP trained here had 3 hidden layers, one hidden layer had 128 neurons, second hidden layer had 64 neurons and last hidden layer had 32 neurons with 3 neurons for output and 6000 TFIDF feature vector as input. The F1 score for the model trained to classify between the 3 classes was 0.45. So the overall F1 score is taken as average which is around 0.6. Confusion Matrix:



7 Results and Analysis

Experiment 1 did not yield very good results with the maximum f1 score being 0.38 on test data using MLP. Experiment 2 however gave a much better result. We obtained a score of 8076 using the scoring metric used in the fake news challenge.

8 References

1. S. M. Padnekar, G. S. Kumar and P. Deepak, "BiLSTM-Autoencoder Architecture for Stance Prediction," 2020 International Conference on Data Science and Engineering (ICDSE), 2020, pp. 1-5, doi: 10.1109/ICDSE50459.2020.9310133.

2. M. Umer, Z. Imtiaz, S. Ullah, A. Mehmood, G. S. Choi and B. -W. On, "Fake News Stance Detection Using Deep Learning Architecture (CNN-LSTM)," in IEEE Access, vol. 8, pp. 156695-156706, 2020, doi: 10.1109/ACCESS.2020.3019735.

3. Thota, A., Tilak, P., Ahluwalia, S., Lohia, N. (2018). Fake news detection: a deep learning approach. SMU Data Science Review, 1(3), 10.