Introduction

Nowadays, Internet is considered as the most important source of opinions coming from billions of people spread across the world. These opinions are a really valuable source of feedback and various other information about current issues, events, products, services, etc. These huge loads of opinions coming from all over the Web provides an exceptional opportunity for decision makers to deploy these sources of data to identify public opinions about their target of interest and beliefs.

One such instance is how, modern policy makers who are interested in determining citizens reaction to new policies may use posts from social media platforms like Facebook and Twitter related to their target policy in order to collect feedback and make appropriate changes as per need. The main challenge is the tremendous amount of such textual data that makes their manual analysis time-consuming, costly and to a great extent, impossible. Hence, need of Automation of such tedious process arises and efforts for automatic text understanding and analysis are part of the Natural Language Processing research area. The majority of NLP methods are based on statistical methods inspired from various fields such as Machine Learning, Data Mining, Information Retrieval and Deep Learning.

Sentiment analysis and Stance detection can be considered as transitional steps towards understanding textual data. We believe that stance classification goes one step further than sentiment analysis as it is more complicated than classifying text into expressing a positive or a negative opinion. Stance detection is a subcategory of opinion mining, where the task is to automatically determine whether the author of a piece of text is in favour or against a given target. Stance detection is related to, but not the same as sentiment analysis. In sentiment analysis, we are interested in whether a piece of text is positive, negative, or neutral based on just the content of the language used. Typically, for sentiment analysis, the choice of positive or negative language correlates with the overall sentiment of the text. However, the stance of a piece of text is defined with respect to a target topic, and can be independent of whether positive or negative language was used. The target (topic towards which opinion is expressed) may or may not be mentioned directly in the actual text, and any entities mentioned in the text may or may not be the actual target of opinion.

Our main objective is to explore stance detection in social media texts, particularly tweets and news comments. Automatically detecting stance has widespread applications in information extraction, text summarization, and textual entailment. Recently, stance detection has been widely used in fake news detection. Specifically, the task is formulated as detecting the stance toward the news headline for different text spans of the news article to detect contradictions.

Background Study

Significant progress has been made in the field of Stance analysis and great amount of research work has been done to achieve considerable success in this area. It wasn't possible to get such extensive work covered up under such a limited time for this study, so we have picked a few research papers based on our understanding of their credibility and relation to our work. A brief summary of three out of them is presented here:

Paper 1 – Stance and Sentiment in Tweets

Authors – Saif M. Mohammad, Parinaz Sobhani, Svetlana Kiritchenko

Publication Year – 2016

- **Problem Statement –** The task that is explored here is formulated as follows: given a tweet text and a target entity (person, organization, issue, etc.), automatic natural language systems must determine whether the tweeter is in favour of the given target, against the given target, or whether neither inference is likely.

- **Motivation –** Although, access to both stance and sentiment annotations allows to explore several things. Its shown that although knowing the sentiment expressed by a tweet is beneficial for stance classification, it alone is not sufficient, hence more things need to be figured out.

- **Methodology –** A linear-kernel support vector machine (SVM) classifier that relies on features drawn from the training instances, such as word and character n-grams, as well as those obtained using external resources, such as word-embedding features from additional unlabelled data is used to classify stance.

- **Dataset –** Under this work, the first dataset of tweets labelled for both stance and sentiment is created. More than 4,000 tweets are annotated for whether one can deduce favourable or unfavourable stance toward one of five targets: "Atheism," "Climate Change is a Real Concern," "Feminist Movement," "Hillary Clinton," and "Legalization of Abortion." Each of these tweets is also annotated for whether the target of opinion expressed in the tweet is the same as the given target of interest. Finally, each tweet is annotated for whether it conveys positive, negative, or neutral sentiment.

- **Results –** The proposed model is simple but effective stance detection system that obtained an F-score of 70.3, higher than the one obtained by the more complex models.

Paper 2 – Automatic Stance Detection Using End-to-End Memory Networks

Authors – Mitra Mohtarami, Ramy Baly, James Glass, Preslav Nakov, Lluís Màrquez, A. Moschitti

Publication Year – 2018

- **Problem statement** – Creation of a novel end-to-end memory network for stance detection, which jointly (i) predicts whether a document agrees, disagrees, discusses or is unrelated with respect to a given target claim, and also (ii) extracts snippets of evidence for that prediction.

- **Motivation** – An unprecedented amount of false information has been flooding the Internet with aims ranging from affecting individual people's beliefs and decisions to influencing major events such as political elections. Consequently, manual fact checking has emerged with the promise to support accurate and unbiased analysis of public statements. As manual fact checking is a very tedious task, automatic fact checking has been proposed as an alternative.

- **Methodology** – A novel memory network model enhanced with CNN and LSTM networks for stance detection is being utilised. Further a novel extension of the general architecture is proposed based on a similarity-based matrix, which is used at inference time, and showed that this extension offers sizable performance gains. Finally, it is shown that this model is capable of extracting meaningful snippets from the input text document, which is useful not only for stance detection, but more importantly can be useful for human experts who need to decide on the factuality of a given claim.

- **Dataset** – The dataset provided by the Fake News Challenge is used, where each example consists of a claim–document pair with the following possible relationship: agree, disagree, discuss, unrelated. The data includes a total of 75.4K claim document pairs, which link 2.5K unique articles with 2.5K unique claims, i.e., each claim is associated with 29.8 articles on average.

- **Results** – The Gradient Boosting classifier gave around 75% weighted accuracy and the advanced Neural Network used gave 81% weighted accuracy.

<u>Paper 3</u> – MITRE at SemEval-2016 Task 6: Transfer Learning for Stance Detection

Authors – Guido Zarrella, Amy Marsh

Publication Year – 2016

- **Problem statement** – Making a system for performing automatic stance detection in social media messages.

- **Motivation** – To the human observer messages like these contain an interpretable stance relevant to the topic of climate change. But to understand rhetorical devices like sarcasm, irony, analogy, and metaphor, a reader often uses personal experience to infer broader context. For machines, matters are additionally complicated by use of informal vocabulary, grammar, and spelling. Furthermore, training data is often expensive or difficult to collect in bulk. These challenges motivated our efforts to seek transfer learning of broad world knowledge through feature pre-training using large unlabelled datasets.

- **Methodology** – A recurrent neural network initialized with features learned via distant supervision on two large unlabelled datasets was employed. We trained embeddings of words and phrases with the word2vec skip-gram method, then used those features to learn sentence representations via a hashtag prediction auxiliary task. These sentence vectors were then finetuned for stance detection on several hundred labelled examples.

- **Dataset** – This approach was able to maximize the value of limited training data by transferring features from other systems trained on large, unlabelled datasets. Though, transfer learning does not completely eliminate the need for labelled data in domain of training data.

- **Result** – This submission achieved an average F1 score of 67.8 on the FAVOR and AGAINST classes of the held-out test set, which contained tweets from all five topics. This was the top scoring system among the 19 entries submitted to the supervised stance detection shared task.

Paper 4 – Stance Detection with Hierarchical Attention Network

Authors – Qingying Sun, Zhongqing Wang, Qiaoming Zhu and Guodong Zhou

Publication Year – 2019

- **Problem statement** – Aim is to focus on stance detection task considering the impacts of different linguistic features.

- **Motivation** - Most of the previous works, model the sequence of words to learn document representation. However, much linguistic information, such as polarity and arguments of the document, is correlated with the stance of the document, and can inspire us to explore the stance. In addition, since the influences of different linguistic information are different, we propose a hierarchical attention network to weigh the importance of various linguistic information, and learn the mutual attention between the document and the linguistic information.

- **Methodology** - A hierarchical attention model is proposed, which stands for the mutual attention between the document and the linguistic factors. The model contains two parts: linguistic attention part and hyper attention part. The former helps learn flexible and adequate document representation with different linguistic feature set, and the latter helps adjust the weight of different feature sets.

- **Dataset** - Two datasets are used to evaluate the performance of the proposed system. H&N14 is collected by Hasan and Ng, and SemEval16 is from SemEval-2016 Share Task. H&N14 is collected from an English online debate forum with four targets: “Abortion”, “Gay Rights”, “Obama”, and “Marijuana”. SemEval16 is the dataset for stance detection from English tweets, and each tweet corresponds to a special target: “Atheism”, “Climate Change is a Real Concern” (“Climate”), “Feminist Movement” (“Feminist”), “Hillary Clinton” (“Hillary”), and “Legalization of Abortion” (“Abortion”).

- **Result** - HAN is the proposed model, which uses Hierarchical Attention Neural (HAN) model to learn the mutual attention between document and linguistic information. Although models like SVM and LSTM gave almost similar accuracies.

Requirement Analysis

Requirement for Training

- Intel Core i5 processor @ 2.7 GHz
- 8 GB RAM
- 100 GB Hard Disk / SSD
- An operating system capable of running python environment.

Requirement for Testing

- 2 GHz x86 or x64 processor
- An operating system capable of running python environment.
- 4 GB RAM

Functional Requirement

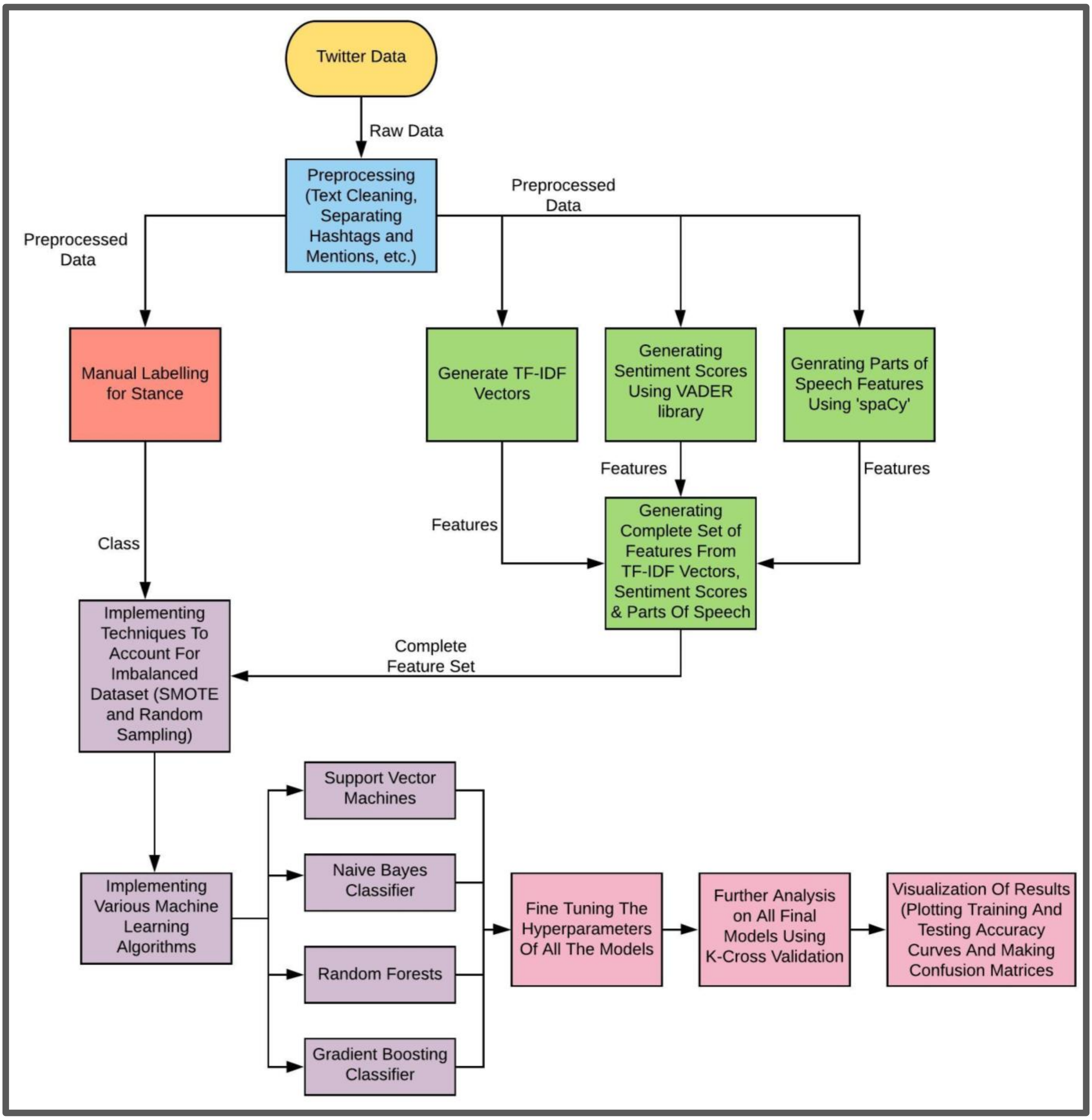
- Social media platform API: To fetch posts that would be required in the dataset.
- Preprocessing Unit: To clean the unstructured text that could be used in models.
- spaCy (POS unit): To generate POS features for further training processes.
- Processing Unit: Extracts TF-IDF features and embedding vectors features.
- Training Unit: Used to train different ML and Neural Networks models.
- Prediction Unit: Used to find training and testing accuracies.

Non-Functional Requirement

- **Reliability** – The system should perform the appropriate calculations and generate alerts in real time, that is highly responsive application is required.
- **High Performance** – The application should be optimized, so that it can run on bare minimum metal.
- **Maintenance** – The application should be maintained so that it is sync with the latest python versions and operating system. Required changes should be made from time to time.

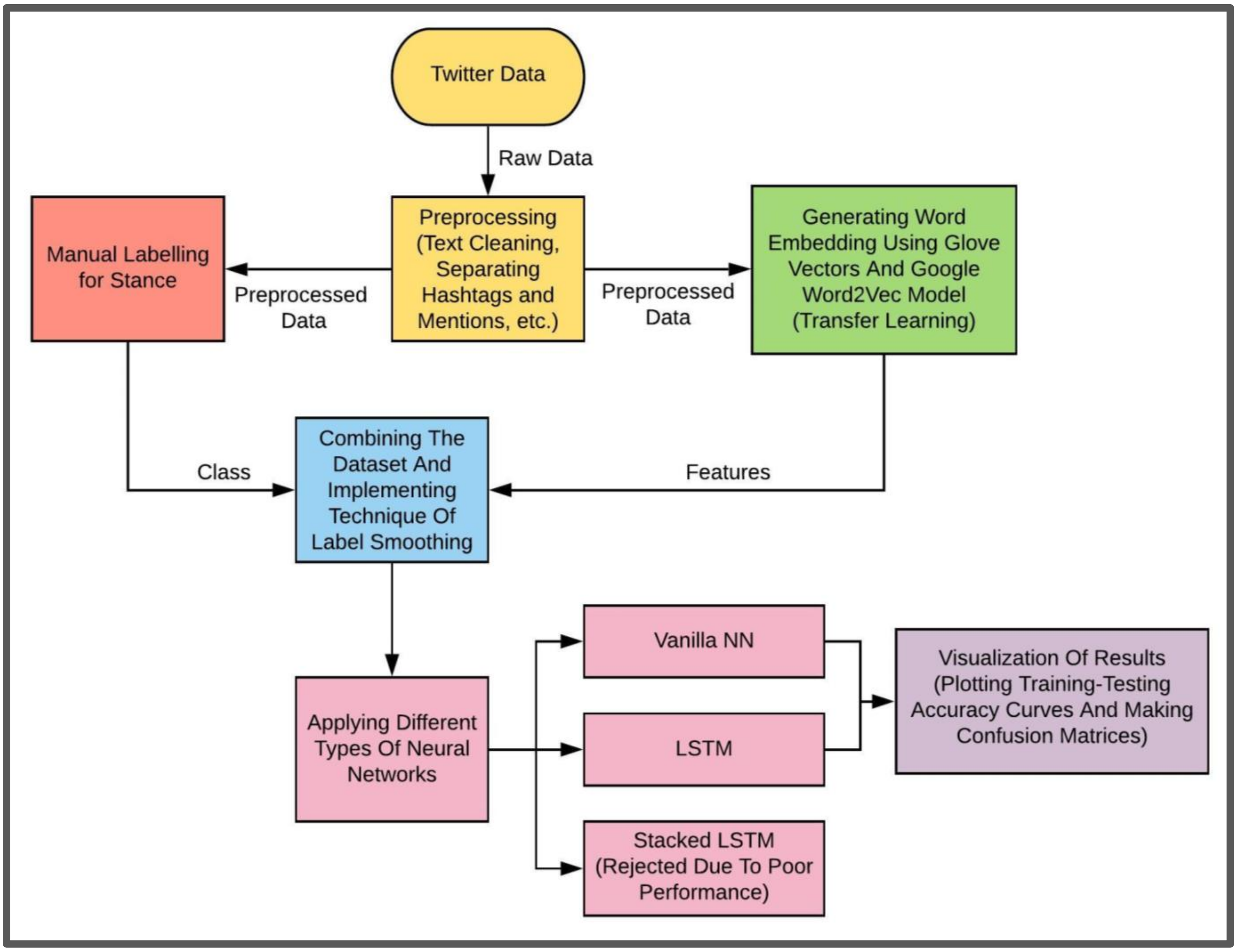
Detailed Design

Detailed Design of the approach using various machine learning algorithms to detect Stance incorporated (1) Collection, (2) Preprocessing, (3) Annotation and Feature Generation, (4) Sampling, (5) Implementation of Algorithms, (6) Analysing and (7) Visualization.



Design Flow Diagram for Machine Learning Approach

Detailed Design of the approach using various Neural Networks to detect Stance incorporated (1) Collection, (2) Preprocessing, (3) Annotation (4) Transfer Learning (5) Implementation of Models, (6) Training and Testing, (7) Analysing and Visualization.



Design Flow Diagram for Neural Network Approach.

Implementation

I. Initial steps were to prepare the dataset for Stance Detection. Twitter posts were collected related to a specific event, General Elections 2019. Text from the posts were preprocessed. We separated hashtags and mentions from the tweets after performing all the basic text cleaning operations. These cleaned tweets were further processed as per the requirement of the model. All posts were labelled for the Stance class in accordance of the standard methods followed while annotating such datasets. Label values were properly scaled and mean results were considered as the final labels. Our supervisor cross-checked these annotations.

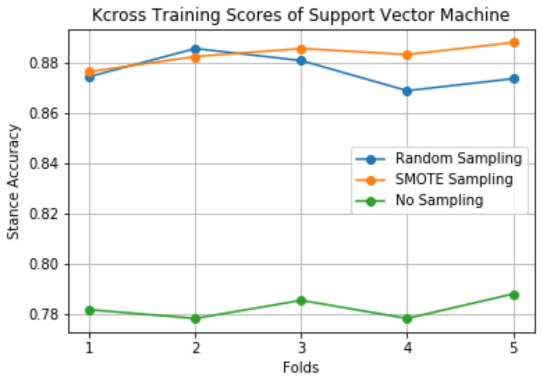
II. We have followed two approaches to analyse the data and generate models that can predict stance for this study. Under first approach we used TF-IDF vectors, sentiment scores and POS as features and we applied multiple Machine Learning algorithms and sought to find the best working models among them. Other way around, we used various Neural Networks on the textual data with the help of Transfer Learning. A brief description for the following two approaches is as follows –

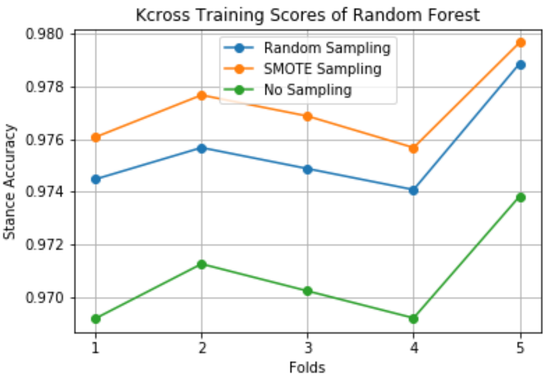
A. To prepare the feature set for the Machine Learning models, after some research we finalized three things, first TF-IDF vectors, second POS features and last being sentiment scores. After some more pre-processing we also performed lemmatization on the text. POS features were measured for each post using 'spaCy' library and stored into the dataset. We also calculated sentiment scores using VADER library. Further we used TF-IDF vectorizer and added these features to complete our dataset for the approach. We also used techniques like SMOTE Sampling and Random Sampling to account for the problem of imbalanced dataset. After this, we fed this dataset to various Machine Learning algorithms like Gaussian Naïve Bayes, Random Forests, SVM and Gradient Boosting Classifier using transforms like Binary Relevance, Classifier Chains and Power Set from 'scikit-multilearn'. After fine tuning the hyper-parameters we finalized a model from all the four algorithms and then we performed K-Cross validation on all the four algorithms and further improved our models based on the results. Then we plotted training and validation accuracy curves. We completed our analysis by visualizing the results further by plotting Confusion Matrices and then performed a comparative study among the four models.

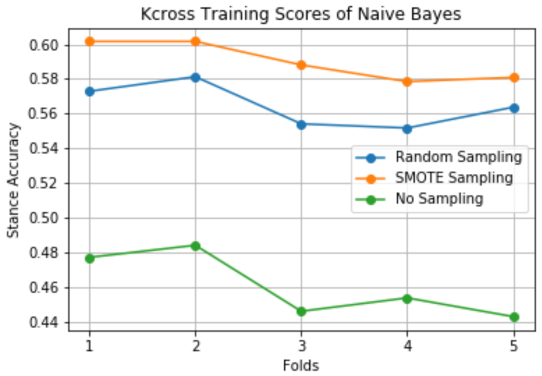
B. Neural Networks are known to perform better in tasks related to NLP and are considered as the next step after the Machine Learning algorithms. So, after experimenting enough with the ML models we started implementing Neural Networks for our task of stance detection. We tried different things that came up as appealing during our literature survey and implemented three different Neural Networks. First, we started with a Vanilla Neural Network, though it is the most basic model, but it performed quite well on our dataset. We tuned the hyperparameters of the Vanilla NN model and visualized the results. Our second model was a basic LSTM model, we used Transfer learning along with the LSTM by making word embeddings as per our need (using GoogleNews-Vector). We also implemented a Stacked LSTM model which have an extra layer of LSTM cells, in this we used the Twitter Glove vector for transfer learning. After implementing the technique of Label Smoothing for accounting the problem of Imbalanced dataset and hyperparameter tuning we found that Stacked LSTM's performance was quite poor as compared to the basic LSTM and hence we moved forward with the basic LSTM as our final LSTM model. We visualized the results of the final model and compared both Vanilla NN and LSTM model as well. Although both have nearly same performance, LSTM had slightly better accuracies.

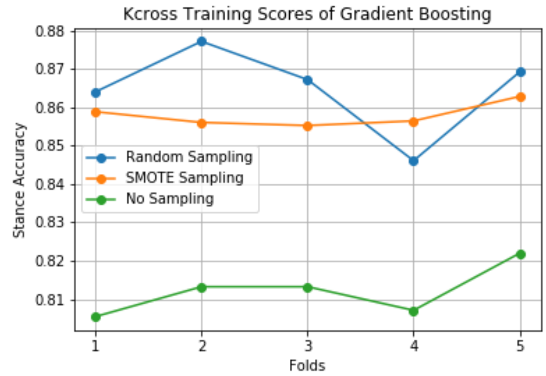
Experimental Results and Analysis

1. K-Cross Training Score Graphs of All Machine Learning Models

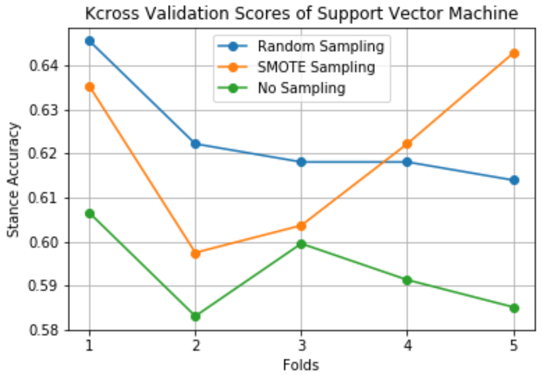


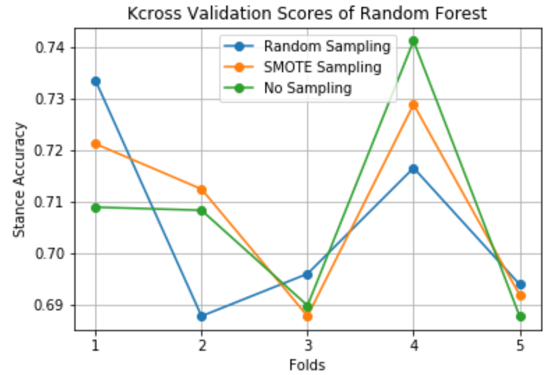


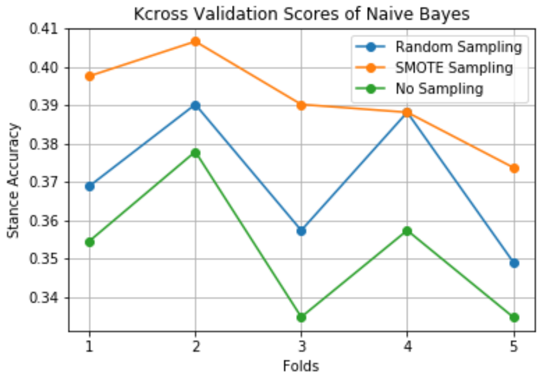


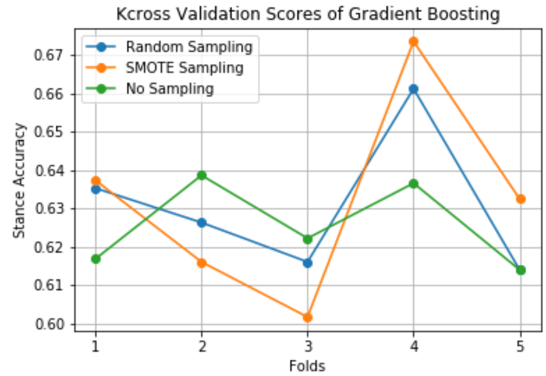


2. K-Cross Validation Score Graphs of All Machine Learning Models

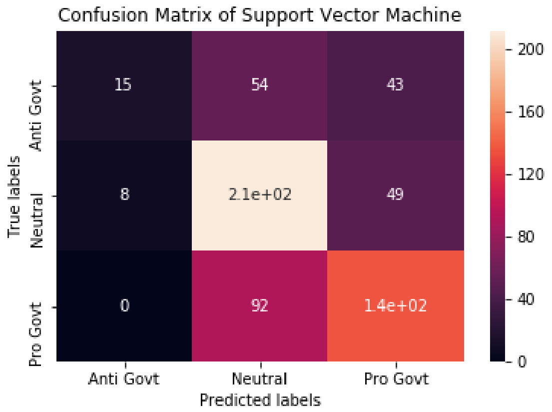


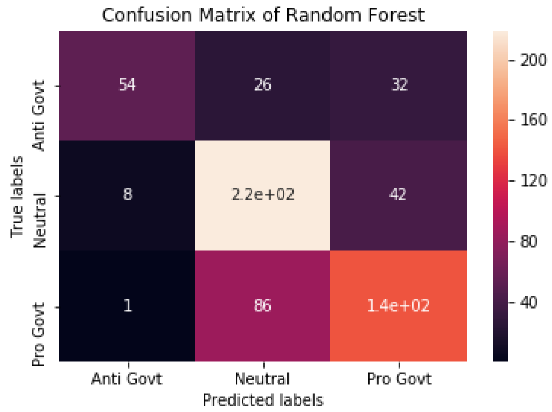


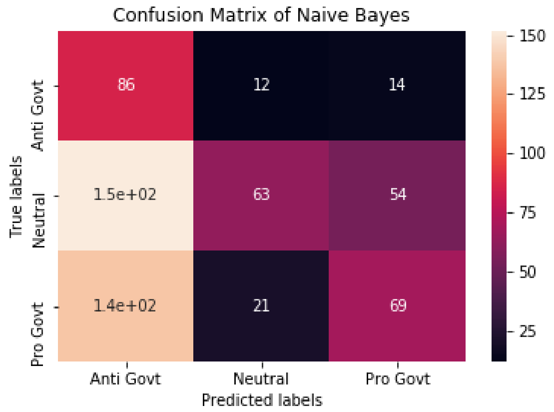


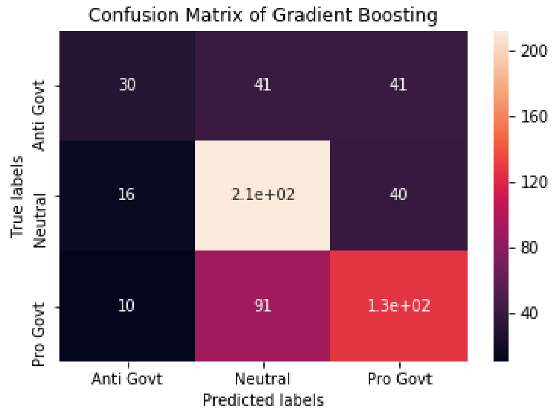


3. Confusion Matrix of All Machine Learning Models

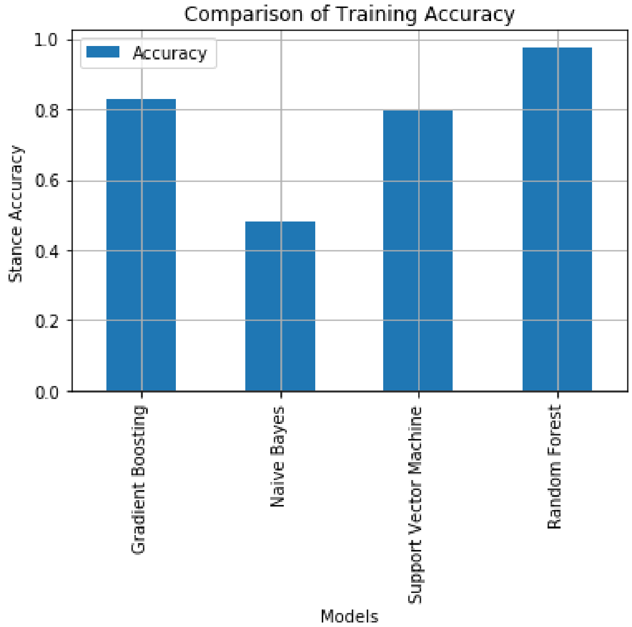


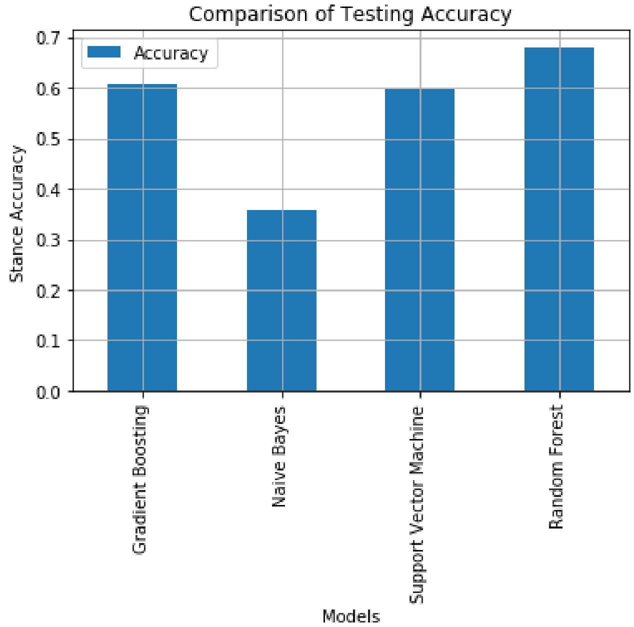




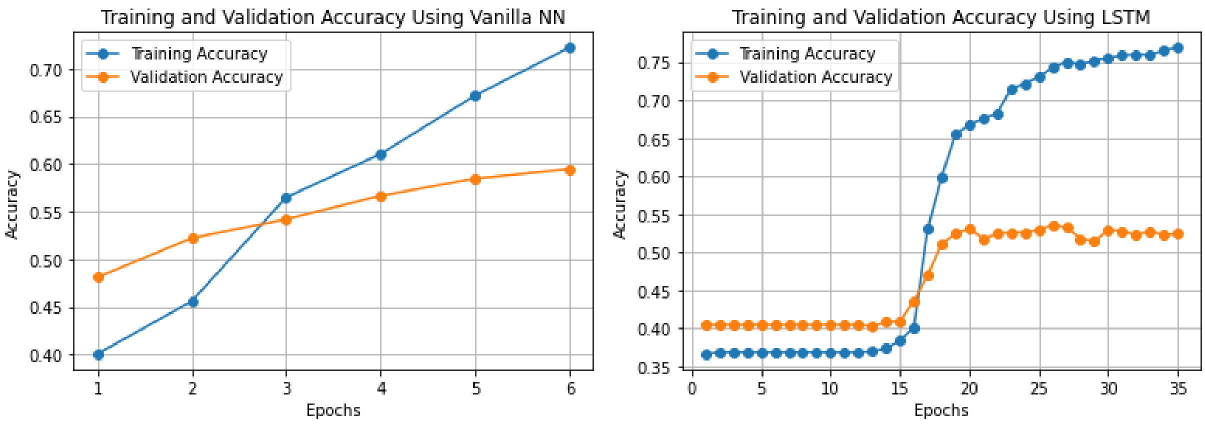


4. Comparing Training and Testing Accuracies of All Machine Learning Models

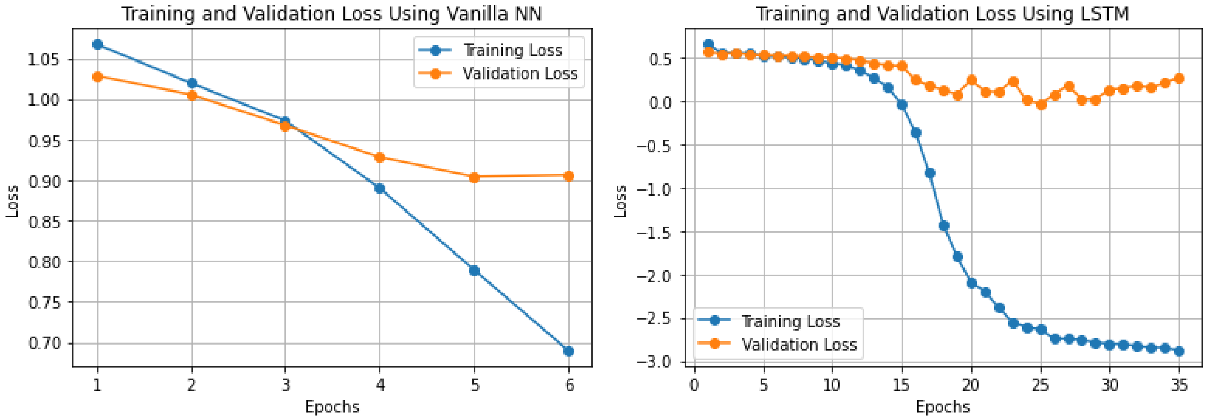




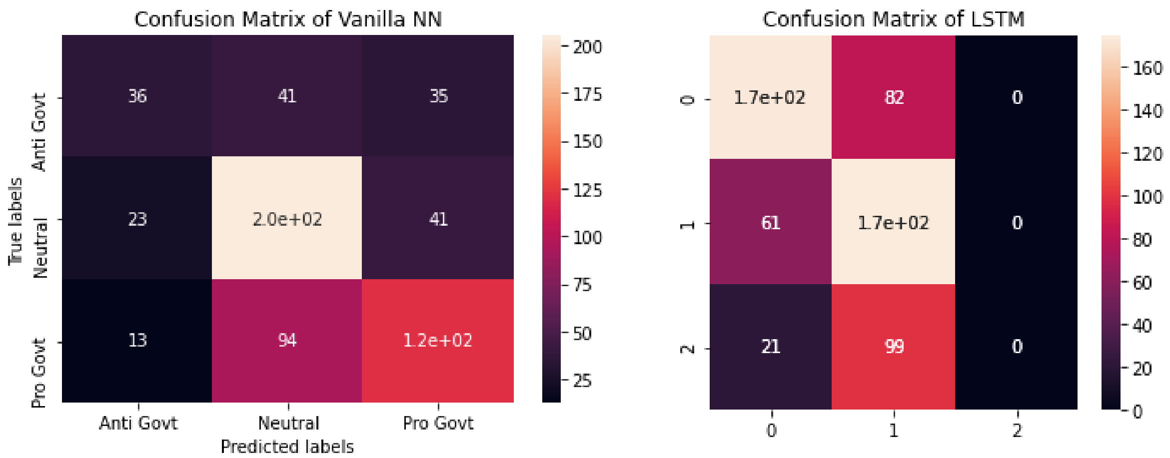
5. Training and Validation Accuracy of Neural Networks



6. Training and Validation Loss of Neutral Networks



7. Confusion Matrix of Neural Networks



8. Summary of All Models

Accuracies of all the final models are tabulated here.

Model	Training Accuracy	Validation Accuracy
SVM	88.30%	62.06%
Random Forest	97.71%	70.85%
Naïve Bayes	59.01%	39.12%
Gradient Boosting	85.78%	63.21%
Vanilla NN	72.19%	59.44%
LSTM	76.48%	56.89%

Conclusion

After researching extensively about the work related to the problem of Stance detection and rigorously working to implement the stuff that seemed to better for our work, we made several models that gave comparable accuracies. To conclude this work, we want to put forth these points:

- We created our own dataset by picking the target as per our own choice that was General Elections 2019, we collected the data from Twitter and after performing all the required pre-processing we generated the features like TF-IDF, POS and sentiment scores to compile a dataset which according to us now have good range of features for moving forward with this work. We also labelled our data following the standard rules followed in such tasks. And hence we were able to present a dataset that to our belief can be used in another research works as well.

- As our dataset was little imbalanced, we found out that techniques like SMOTE and Random sampling can solve this problem in ML based models and Label Smoothing in Neural Networks to a great extent.

- Various ML algorithms were implemented throughout the course of work and the models were tuned extensively. This gave us some well performing models though there was problem of Overfitting that we tried to minimize as much as possible.

- Transfer Learning proved to be a promising method as per our research and work as well. We tried to use this extensively in our work and there was clear improvement in accuracies.

- Vanilla NN and Basic LSTM models performed at par with the ML models although they failed to surpass the results of ML models which was expected. This was due to the fact that we had a limited dataset and we believe if more annotated tweets can be added to this dataset our Neural Networks based models would certainly perform better.

- Due to some limitations of the dataset, more complex Neural Networks like Stacked LSTM failed and hence it's clear that some different type of Neural Networks needs to be used in this type of problem in further work.

- In further work we would like to extend our dataset by adding more annotated tweets and adding more features to it. We would also try out more types of Neural Networks that could help to improve the performance.

References

Research Papers

[1] Saif M. Mohammad, Parinaz Sobhani, and Svetlana Kiritchenko. “Stance and Sentiment in Tweets.”; 2017.

[2] Mitra Mohtarami, Ramy Baly, James Glass, Preslav Nakov, Lluís Màrquez, A. Moschitti; “Automatic Stance Detection Using End-to-End Memory Networks”; 2018.

[3] Guido Zarrella, Amy Marsh; “MITRE at SemEval-2016 Task 6: Transfer Learning for Stance Detection”; 2016.

[4] Qingying Sun, Zhongqing Wang, Qiaoming Zhu and Guodong Zhou; “Stance Detection with Hierarchical Attention Network”; 2019.

[5] Kuntal Dey, Ritvik Shrivastava, Saroj Kaushik; “Topical Stance Detection for Twitter: A Two-Phase LSTM Model Using Attention”; 2018.

[6] Parinaz Sobhani; “Stance Detection and Analysis in Social Media”; 2017.

[7] Vasiliki Simaki, Carita Paradis, and Andreas Kerren; “Stance Classification in Texts from Blogs on the 2016 British Referendum”; 2017.

[8] Rui Dong, Yizhou Sun, Lu Wang; “Weakly-Guided User Stance Prediction via Joint Modeling of Content and Social Interaction**”; 2018.**

[9] Jiachen Du, Ruifeng Xu, Yulan He, Lin Gui; “Stance Classification with Target-Specific Neural Attention Networks**”; 2019.**