COL865 (Social Computing) : Project Milestone 1

Predicting Election Results using Sentiment Analysis on Tweets
and Detecting Bias in Media Coverage

Navya Jain (2019CH10106)[1], Kshitij Alwadhi (2019EE10577)[2], and Bhavuk Bhandula (2019MT10683)[3]

[1]Department of Chemical Engineering, IIT Delhi
[2]Department of Electrical Engineering, IIT Delhi
[3]Department of Mathematics, IIT Delhi

5th February 2022

Contents

1 Problem Statement

The prime aim of this project would be to study the correlation between sentiments predicted from the tweets of the users before the polling period and the election results. We also plan on studying if we can analyse the tweets in each sentiment and extract the topics which trigger them, i.e. if there are specific categories that initiate a positive or negative sentiment towards a political party. These topics can be agriculture, price hike etc which can be very useful to filter out topics that drive public interest and also for the country's development. Another direction we plan on taking is to study the bias in the coverage related to political parties done by the Indian media twitter accounts.

2 Approach

We would be following a three pipeline approach in our project. We cover a brief overview of these approaches in the first milestone and we would explore them in detail in the subsequent milestones.

2.1 First Pipeline

In the first stage of this project, we will train a sentiment analysis model on the Sentiment140 Dataset. We plan on trying two models here, a TF-IDF based SVM model and a Gensim + Keras based model. We will use a twitter dataset from some past elections to validate the performance of this model and bring about a mapping from the overall sentiment predicted to the relevant metrics related to elections. A block diagram explaining this approach is given in Figure 1

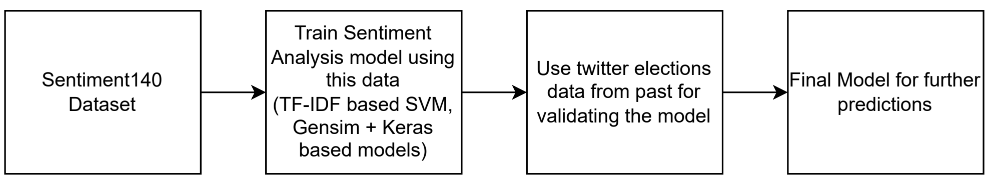

Figure 1: First Pipeline

2.2 Second Pipeline

In the second stage, we will use this model for making predictions on the custom dataset that we will scrape ourselves from Twitter related to the upcoming elections in the states of Uttar Pradesh and Punjab. We also plan on studying if we can analyse the tweets in each sentiment and see if we can extract the topics which trigger them. A block diagram explaining this approach is given in Figure 2

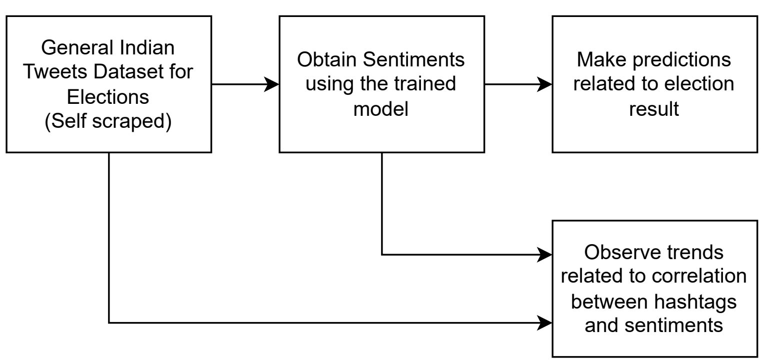

Figure 2: Second Pipeline

2.3 Third Pipeline

Another approach that we plan on pursuing is to use this trained model on a similar dataset scraped by us concerning with the tweets made by Indian media twitter accounts related to political parties participating in the elections. We plan on studying if there is a bias of sentiment towards a particular party by these media channels. A block diagram explaining this approach is given in Figure 3

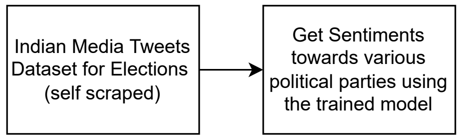

Figure 3: Third Pipeline

3 Literature Review

3.1 Analyzing Delhi Assembly Election 2015 Using Textual Content of Social Network

3.1.1 Abstract

In this paper[1], the authors have performed an analysis of the textual content of Twitter's data related to Delhi Assembly Election 2015 to predict election results. The main contributions of this paper include preparation of the dataset and designing mapping function to map the Twitter's sentiment share to seat counts of top 3 contesting parties, with minimum RSME. It was observed that the results were very close to ground reality.

3.1.2 Dataset

Tweets related to Delhi Assembly Election 2015 were collected using Twitter's search API [2]. It was collected for a period of 31 days, considering this period as a peak period for election campaigning. The data collection was restricted to only the 3 major contesting parties : BJP, AAP, Congress and a search list was manually curated comprising popular words related to the party such as star campaigners and derivative terminologies. For pre-processing the tweets, 1) all duplicate tweets were removed, if re-tweeted by same user ID and 2) all non-English and multilingual tweets were also discarded from the dataset. After all pre-processing techniques, they curated 3,52,730 tweets.

3.1.3 Approach

The pipeline of the approach of the paper is presented in 4. Briefly discussing the steps involved :

1. **Preparation of Training Dataset** : Three datasets were used to curate the training dataset for sentiment analysis, i.e. an IMDb training dataset, Manually annotated tweets from Delhi Assembly Election 2015 and emoticons ":)" and ":(" to classify tweets. After processing these datasets, positive and negative polarity tweets are segregated while neutral are discarded.

2. **Sentiment Analysis of Twitter Data to Infer Actual Vote Share** : Two approaches have been discussed :-

   (a) *Mere count of tweets mentioning a party or candidate reflects the election results. The number of tweets is directly proportional to the vote rate* : This approach [3] was used primitively and has faced severe criticism [4] recently, simply because a political party might be talked about a lot on social media, not because of it's followers but for the negative politics, which will not be reflected in the election results. So, simply a count of tweets mentioning a party does not suffice.

(b) *The ratio of total number of positive tweets belongs to a party P with the sum of the total number positive tweets of all parties, is proportional to the real vote share of the party P* : This approach has been used in the paper, which uses sentiment analysis to infer real vote share. Precisely it follows that :

$$\frac{Total\ Number\ of\ Positive\ tweets\ belonging\ to\ a\ party\ P}{Total\ positive\ tweets\ of\ all\ parties} \propto (Vote\ Share\ of\ a\ party\ P)$$

3. **Sentiment Analysis** : This is a 2-tier process, firstly checking the subjectivity of a tweet i.e. if it represents an opinion or not using the opinion word list SENTIWORDNET and segregating tweets by assosciating each tweet on the basis of the search term to one of the three sets (BJP,AAP,INC). Secondly, a sentiment analysis is applied to determine polarity of each tweet using SVM or WEKA.

4. **Modeling Mapping Functions to convert Sentiment Share to Seat Share** : To convert the sentiment share to a seat share, a model was designed to determine seat share using sentiment share. The model was based purely on statiscal data of previously held elections in India to predict seat share of top 3 parties with some acceptable error.

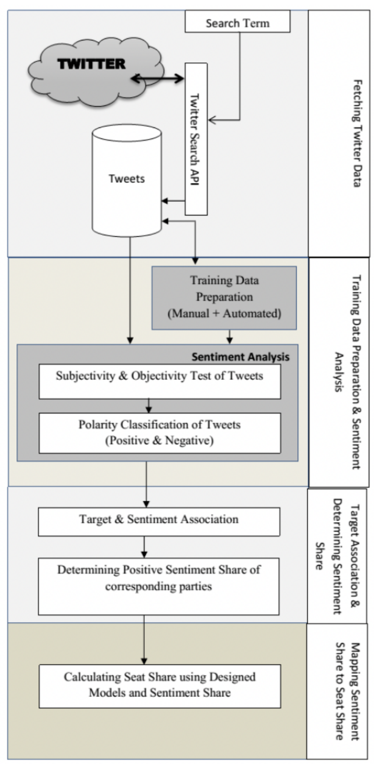

Figure 4: Proposed Approach

3.1.4 Results

1. **Predicting Number of Seats Using The Model** : The results obtained in the Delhi Assembly Election 2015 calculated using the model and the actual results are presented in 5, which seems very close to actual results.

2. **Accuracy of Sentiment Analysis and Its Impact to Seat Share**: A SVM with 10-fold cross validation was used with an accuracy of 93.7%. VCR is defined as the ration of the total

Table 3. Predicting Number of Seats Using Our Model

Rank of parties	Party Name	Actual Vote Share	Calculated PSS in (%)	Actual Seat Distribution	Calculated Seat Distribution
1st party	AAP	54.3%	53.48%	67	A=67.4
2nd Party	BJP	32.2%	34.8%	03	B=1.8
3rd Party	INC	9.7%	11.72%	00	C=0.71

Figure 5: Results

number of positive tweets belonging to a party P to the total number of seats won by a party P. As per the model, the VCR was 1245, 32012, 21433 for AAP, BJP and INC.

3.1.5 Takeaways

Analysing the contents of this paper, we understand that there is an extreme potential of Twitter's Data in the prediction of election results since virtual communities today resemble closely to real communities. Secondly, it is important to design a function using statistical data from previously held elections to predict seat share from sentiment share. Further, having a training dataset to accurately grasp event specific and temporal terminologies should be taken care of as well.

3.2 Twitter Sentiment Analysis: The Good the Bad and the OMG!

3.2.1 Abstract

In this paper[5], the author investigates the utility of linguistic features for detecting the sentiment of Twitter messages. They extended the existing lexical resources with features that capture information about the informal and creative language used in microblogging, which is common in social media platforms.

3.2.2 Dataset

The author obtained 3 corpora of Twitter message:

1. hashtagged data set (HASH): compiled from the Edinburgh Twitter corpus. Hashtags were identified, and chosen in such a manner that they would be most useful for identifying positive, negative and neutral tweets
2. Emoticon data set (EMOT) : from http://twittersentiment.appspot.com, a now inactive domain. It contains tweets containing emoticons, which used to associate positive or negative sentiment to a tweet
3. A manually annotated data set produced by the iSieve Corporation(ISIEVE). As it is a hand annotated dataset, it was used for evaluation

3.2.3 Approach

The first set of features considered from a tweet are the tweet's unigrams and bigrams. Next, features representing information from a sentiment lexicon as well as those relating to POS were added. Finally, features capturing some of the more domain-specific language of microblogging were considered:

1. n-gram features: First, rudimentary negation was done by attaching the the word not to a word that precedes or follows a negation term. Then, all unigrams and bigrams are identified in the training data and ranked according to their information gain, measured using Chi-squared

2. Lexicon features: Words listed the MPQA subjectivity lexicon are tagged with their prior polarity: positive, negative, or neutral

3. Part-of-speech features: features for counts of the number of verbs, adverbs, adjectives, nouns, and any other parts of speech

4. Micro-blogging features: Binary features that capture the presence of positive, negative, and neutral emoticons and abbreviatons and the presence of intensifiers (e.g., all-caps and character rep-etitions)

Combinations of these features were used to train 4 AdaBoost models:(i) n-grams and lexicon features (n-gram+lex), (ii) n-grams and part-of-speech features(n-gram+POS), (iii) n-grams, lexicon features and microblogging features (n-grams+lex+twit), and finally (iv) all the features combined.

3.2.4 Results

The best performance on the evaluation datacomes from using the n-grams together with the lexicon features and the microblogging features. Including the part-of-speech features actually gives a drop in performance. This is showcased in figure 6

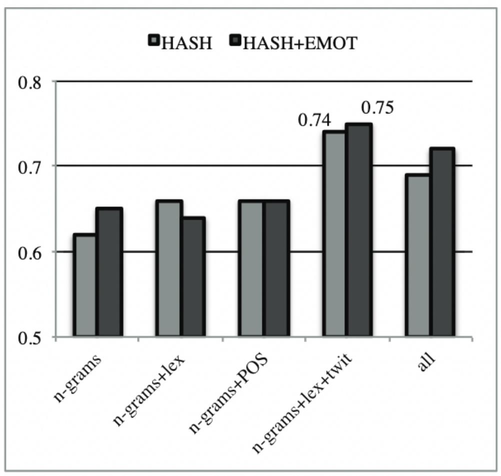

Figure 6: Average accuracy on test set (”twit” refers to microblogging features)

3.2.5 Takeaways

Through the paper, we realise that including informal linguistic features can improve the accuracy of social media sentiment analysis. Microblogging caused a drastic increase in accurary, and therefore we plan on using its ideas in our classification process.

3.3 Sentiment Analysis and Sarcasm Detection of Indian General Election Tweets

3.3.1 Abstract

In this paper[6], the authors aim at analysing the sentiments of the people of India during the Lok Sabha Election of 2019 using the twitter data of that duration. Unlike the general approach, this paper discusses using a Transfer Learning Technique to cater the unsupervised nature of the dataset. Further, Linear Support Vector Classifiers and a TF-IDF approach is used for handling the textual data of tweets. This paper also discusses an interesting look at addressing the sarcastic tweets posted by users, which is not researched upon in depth yet.

3.3.2 Dataset

The India Lok Sabha Elections-2019 tweets have been used to test the trained model. The complete dataset is available at [7].

3.3.3 Approach

The proposed model works in two levels, pictorially represented in 7 and 8.

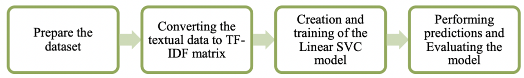

Figure 7: The flow of work for the training of sentiment analysis and sarcasm detection model

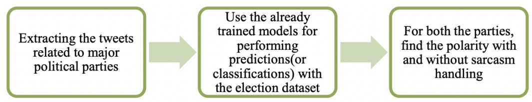

Figure 8: The flow of work for the testing of the trained models on the election's dataset

The authors first used the standard Twitter review dataset available on Kaggle [7] for training the Linear SVC model and then used this model to perform sentiment analysis on the actual election data.

3.3.4 Results

The authors achieve an accuracy of 80% in sentiment analysis model and a 84% accuracy in sarcasm detection model. The positive polarity found with and without sarcasm detection is presented in 9.

Party	Positive Polarity of tweets (without sarcasm handling)	Positive Polarity of tweets (with sarcasm handling)	Negative Polarity of tweets (without sarcasm handling)	Negative Polarity of tweets (with sarcasm handling)
BJP	25.58%	23.76%	13.16%	20.01%
INC	6.50%	6.34%	4.88%	6.64%

Figure 9: Polarity Percentage of tweets with respect to total tweets

3.3.5 Takeaways

From this paper, we discover a new approach for handling the sentiment analysis for a large unsupervised dataset, i.e. the use of transfer learning, along with the handling of sarcastic tweets and analysing tweets for positive and negative polarities. Even though, the paper was able to achieve close to reality results it points out a few reasons which might cause the vote percentage difference between the model's results and the actual election results. These include the digital divide problem, as the entire population does not have access to Internet and even if they do, not everyone uses twitter. Also, due to many political and personal reasons a lot of people avoid sharing negative thoughts on public platforms.

3.4 Sentiment Analysis on Twitter using Neural Network: Indonesian Presidential Election 2019 Dataset

3.4.1 Abstract

In this paper[8], the authors aimed at classifying the sentiment on Indian Presidential Election 2019 tweets dataset. They tried out various architectures such as a Convolutional Neural Network (CNN)[9], Long Short term Memory (LSTM)[10], CNN-LSTM[11], GRU[12] and Bidirectional LSTM[13]. They also compared the performance with traditional ML approaches.

3.4.2 Dataset

They collected the data across two time periods, one before the presidential elections and one after the elections. They crawled the data 5 times before the elections and they chose to crawl the tweets the next day after televised debates between the presidential or vice-presidential candidates were aired. They used a pseudo labelling approach to label their data. In this, they used a pre-annotated dataset of tweets (1369 tweets in number) and trained a Naive Bayes model for sentiment analysis. They then used this model for assigning labels to the rest of the dataset. Their pipeline is shown below in Figure 10.

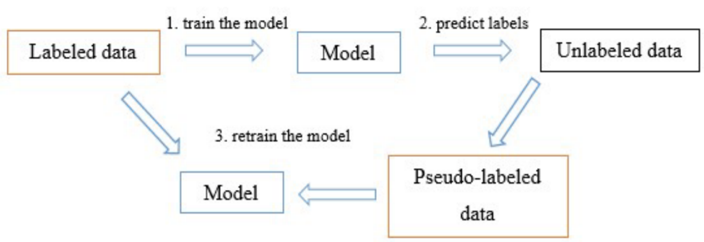

Figure 10: Pseudo-labelling

3.4.3 Approach

After pre-processing the tweets using the general techniques such as stemming, hashtag removal etc, they tried out various model architectures for classifying the sentiments of the tweets. They started with simpler models such as SVM, Logistic Regression etc. Then they moved to the newer models such as CNN, LSTM, CNN+LSTM, GRU+LSTM and bi-LSTM.

3.4.4 Results

The metric they used for comparing the performance was Accuracy. The following are the results they got from this.

Method	TF-IDF Feature	Accuracy (%)
SVM	Yes	**84.04**
	No	83.23
Logistic Regression	Yes	83.15
	No	82.74
Multinomial Naïve Bayes	Yes	82.13
	No	82.08

Figure 11: Using traditional approaches

Method	Accuracy(%)
LSTM	84.20
CNN	84.05
CNN+LSTM	84.30
GRU+LSTM	84.50
Bidirectional LSTM	**84.60**

Figure 12: Using modern approaches

3.4.5 Takeaways

The takeaway from this paper is that, using a pre-labelled data for pseudo-labelling an unseen data is also an effective approach. Moreover, as we can see from the results, they obtained, traditional approaches can also give a compareable accuracy with the modern deep learning based approaches. Moreover, for the task of sentiment analysis, using TF-IDF features seems to be giving better results as compared to performing the analysis without them.

3.5 On predicting elections with hybrid topic based sentiment analysis of tweets

3.5.1 Abstract

Previous work relied on explicit mining of public sentiment using lexical and syntactic features. However, underlying implicit word relations and co-occurrences are overlooked. This task becomes even more challenging in case of short length tweets. Therefore, the paper [14] introduces a novel Method: Hybrid Topic Based Sentiment Analysis (HTBSA). First, latent topics were extracted from a rich corpus using Biterm Topic Model, then sentiment of each topic was learnt using pre-existing lexical sources. Finally, sentiment of each tweet is calculated using sentiment orientation and topic weight. Research is done on 2017 Uttar Pradesh legislative elections.

3.5.2 Dataset

Uttar Pradesh has legislative elections for over a month. Sentiment towards parties and leaders may fluctuate heavily. Therefore, data was collected immediately before elections and during elections. Different key words (hash tags and user names) were used to harvest the tweets, wherein, geo tagging is used for key words which are not exclusive to U.P elections. After manual search for hash tags and usernames, a comprehensive list was prepared including party names, party leader names, multiple official election campaign handles and hashtags on Twitter. To apply geo tagging, longitude, latitude and radius are used. Pre-processing involves following steps: converting into lowercase, stemming, white space, punctuations, symbols, numbers and stop word removal using existing stop words list and a customized list.

3.5.3 Approach

First, BTM was performed on the tweets collected to obtain recurring topics. Sentiwordnet is used to calculate the sentiment associated with each topic. Then, the tweet sentiment is calculated by identifying the corresponding topics and weightedly adding the topic sentiments. Finally, 3 methods are used for vote share prediction:

$$VS_1 = \frac{TTV_i}{\sum_{i=1}^{l} TTV_i} \qquad VS_2 = \frac{TPV_i}{\sum_{i=1}^{l} TPV_i} \qquad VS_3 = \frac{TPM_i}{\sum_{i=1}^{l} TPM_i}$$

where i = 1, 2, ... l [l is number of parties]
VS = Vote Share

TTV = Total Twitter Volume
TPV = Total Positive Volume
TPM = Total Positive Magnitude

3.5.4 Results

The following showcases the results of different techniques:

Vote Share Method	True Share	Twitter Volume (eq 3)	Lexicon-Positive Volume (eq 4)	Lexicon-Positive Magnitude (eq 5) Burnap et al.[7]	HTBSA-Positive Volume (eq 4)	HTBSA-Positive Magnitude (eq 5)
BJP	41.7	40.4	45.5	47.8	43.5	45.23
INC+SP	28.2	43.7	46.3	45.7	42.73	41.19
BSP	22.2	15.82	8.1	6.35	13.75	13.08
Others	7.9	0	0	0	0	0
MAE	-	**7.8**	11	12	**8.2**	8.4

Figure 13: Actual Vote Share vs Vote Share Prediction

The HTBSA based prediction outperformed lexical based prediction. On the other hand, the twitter volume gave comparable results. The authors claim that twitter volume as an approach is not completely reliable and may not produce similar results in future predictions. For instance, in case where majority of tweets are negative, this method will produce erratic results.

3.5.5 Takeaways

The paper provides a new strategy for extracting information from certain tweets. General lexical methods are effective but fails to capture the relevance of hot topics/keywords which are prevalent during elections. These topics may reflect the sentiments towards a leader, a scheme, a political promise, etc. So, if we are able to extract such topics and identify their associated sentiments, we can improve our analysis. Algorithms such as BTM are effective in extracting such information.

3.6 Election Bias: Comparing Polls and Twitter in the 2016 U.S. Election

3.6.1 Abstract

This paper [15] discusses the bias of polls towards election predictions, and compares it with election results obtained through a social media pipeline. Although not related directly to our problem statement, we plan on using the approach used for computing bias towards a candidate in our context to compute news media bias.

3.6.2 Dataset

An already retrieved dataset (Littman et al. 2016) containing 100s of millions of tweet ids was used which either had tweets related to the election or the major candidates.

3.6.3 Approach

The analysis of the textual content of the tweets was conducted using the lexicon and rule-based sentiment analysis python library VADER sentiment. For each user, a compounded sentiment probability was computed on the text of each tweet. After computing the probability, the overall sentiment of each user (Hillary Clinton and Donald Trump) was computed as follows :

1. If the tweet mentions a single candidate : add compounded sentiment probability to the user's overall sentiment towards that candidate.

2. If the tweet mentions both candidate : add compounded sentiment probability for the tweet to both of the user's overall sentiments.
3. If the tweet mentions neither candidate : do not update either of user's compounded sentiments.

The vote of the user was defined as whichever candidate received higher sentiment probability.

3.6.4 Results

The authors found that in the 2016 US election, the media was biased against Donald Trump by -2%, while Twitter was more biased with a 3.4% bias in favour for Donald Trump. While polls had a small bias towards Hillary Clinton, Twitter had a slightly larger bias towards Donald Trump.

3.6.5 Takeaways

This paper discusses an interesting approach to look at all the users at an individual level, analyse their tweets and then determine their vote. As for our problem statement, such an approach could possibly be used to measure the bias of a news channel. Precisely, we can analyse all the tweets by a particular media company, calculate the compounded sentiment probability and add it according to the above approach to the media channel's opinion towards a particular political party. This might help us analyse the extent of a media channel's bias towards a political party.

4 Dataset Description

Broadly, we will be using three categories of datasets for this project. One would be a labelled dataset for training the sentiment analysis model. Then, for validating this model we will be using two twitter datasets, one from 2020 US elections and one from 2019 Indian Assembly elections. Finally, we will be testing out the model on self-scraped datasets. We talk about these datasets in detail in the following subsections.

4.1 Training Dataset

For training our sentiment analysis model, we will be using the Sentiment140 dataset[16]. This dataset has 16 million labelled entries with the data fields being: the polarity of the tweet, ID of the tweet, date of the tweet, query which generated the tweet, username of the person who posted the tweet and finally the text of this tweet. For our purpose, we would require the text of the tweet along with the sentiment associated with it. There are a couple more labelled. datasets[17],[18] which we can use for futher fine-tuning this sentiment analysis model.

4.2 Validation Datasets

For validating our model and bringing about some relation with the sentiment and the outcome of the election, we will be using two datasets. One will be from the 2020 US elections tweets dataset[19] and the other would be the 2019 Indian assembly election tweets dataset[7]. In both of these datasets, we have the text of the tweet and the other metadata related to the tweet. For our purpose, we would require only the text of these tweets from which we aim to predict the sentiment and then bring about a mapping of these sentiments to the metrics related to election results.

4.3 Self-scraped Datasets

There will be two types of datasets that we will be scraping. The overall structure of these datasets will be the same, i.e. they will have the tweet text along with the various meta-data that accompanies them. For performing this scraping task, we will be using the Twitter API for collecting the tweets concerning with the search query/user.

4.3.1 General Tweets

Twitter API enables us to scrape the tweets given a search query similar to what is shown on there website when we search something. For curating this dataset, we will be using a couple of queries related to the political parties contesting in these areas (Uttar Pradesh, Punjab). A few examples of these queries would be #BJP, #BJP4UP, Punjab Elections, Akhilesh Yadav etc.

4.3.2 Indian Media Tweets

Twitter API also enables us to scrape the tweets for a given user. For curating this Indian Media Tweets dataset, we will be using some users such as @ANI, @NDTV etc. One thing to note here is that we will have to use keywords for extracting a subset from this dataset which concerns with the political parties. Another thing is that, since the number of these tweets won't be that high as the General Tweets, we will have to use the GoogleTrans[20] API for translating the non-english tweets to english so as to perform sentiment analysis.

References

[1] Ritesh Srivastava, Horesh Kumar, M. P. S Bhatia, and Shruti Jain. "Analyzing Delhi Assembly Election 2015 Using Textual Content of Social Network". In: ICCCT '15 (2015), 78–85. DOI: 10.1145/2818567.2818582. URL: https://doi.org/10.1145/2818567.2818582.

[2] URL: https://developer.twitter.com/en/docs/api-reference-index.

[3] Andranik Tumasjan, Timm Sprenger, Philipp Sandner, and Isabell Welpe. "Predicting Elections with Twitter: What 140 Characters Reveal about Political Sentiment". In: *Proceedings of the International AAAI Conference on Web and Social Media* 4.1 (2010), pp. 178–185. URL: https://ojs.aaai.org/index.php/ICWSM/article/view/14009.

[4] Andreas Jungherr, Pascal Jürgens, and Harald Schoen. "Why the Pirate Party Won the German Election of 2009 or The Trouble With Predictions: A Response to Tumasjan, A., Sprenger, T. O., Sander, P. G., Welpe, I. M. "Predicting Elections With Twitter: What 140 Characters Reveal About Political Sentiment"". In: *Soc. Sci. Comput. Rev.* 30.2 (2012), 229–234. ISSN: 0894-4393. DOI: 10.1177/0894439311404119. URL: https://doi.org/10.1177/0894439311404119.

[5] Efthymios Kouloumpis, Theresa Wilson, and Johanna Moore. "Twitter Sentiment Analysis: The Good the Bad and the OMG!" In: Jan. 2011.

[6] Arpit Khare, Amisha Gangwar, Sudhakar Singh, and Shiv Prakash. *Sentiment Analysis and Sarcasm Detection of Indian General Election Tweets*. 2022. arXiv: 2201.02127 [cs.IR].

[7] *Indian Political Tweets 2019 (Feb to May) — Kaggle*.

[8] Ahmad Fathan Hidayatullah, Siwi Cahyaningtyas, and Anisa Hakim. "Sentiment Analysis on Twitter using Neural Network: Indonesian Presidential Election 2019 Dataset". In: *IOP Conference Series: Materials Science and Engineering* 1077 (Feb. 2021), p. 012001. DOI: 10.1088/1757-899X/1077/1/012001.

[9] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. "ImageNet Classification with Deep Convolutional Neural Networks". In: *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*. NIPS'12. Lake Tahoe, Nevada: Curran Associates Inc., 2012, 1097–1105.

[10] Sepp Hochreiter and Jürgen Schmidhuber. "Long Short-Term Memory". In: *Neural Comput.* 9.8 (1997), 1735–1780. ISSN: 0899-7667. DOI: 10.1162/neco.1997.9.8.1735. URL: https://doi.org/10.1162/neco.1997.9.8.1735.

[11] Tara N. Sainath, Oriol Vinyals, Andrew Senior, and Haşim Sak. "Convolutional, Long Short-Term Memory, fully connected Deep Neural Networks". In: *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2015, pp. 4580–4584. DOI: 10.1109/ICASSP.2015.7178838.

[12] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. *Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling*. 2014. arXiv: 1412.3555 [cs.NE].

[13] M. Schuster and K.K. Paliwal. "Bidirectional recurrent neural networks". In: *IEEE Transactions on Signal Processing* 45.11 (1997), pp. 2673–2681. DOI: 10.1109/78.650093.

[14] Barkha Bansal and Sangeet Srivastava. "On predicting elections with hybrid topic based sentiment analysis of tweets". In: *Procedia Computer Science* 135 (2018). The 3rd International Conference on Computer Science and Computational Intelligence (ICCSCI 2018) : Empowering Smart Technology in Digital Era for a Better Life, pp. 346–353. ISSN: 1877-0509. DOI: https://doi.org/10.1016/j.procs.2018.08.183. URL: https://www.sciencedirect.com/science/article/pii/S1877050918314728.

[15] David Anuta, Josh Churchin, and Jiebo Luo. "Election Bias: Comparing Polls and Twitter in the 2016 U.S. Election". In: (Jan. 2017).

[16] Alec Go, Richa Bhayani, and Lei Huang. "Twitter sentiment classification using distant supervision". In: *CS224N project report, Stanford* 1.12 (2009), p. 2009.

[17] URL: https://www.kaggle.com/shashank1558/preprocessed-twitter-tweets/version/2?select=processedNeutral.csv.

[18] URL: https://www.kaggle.com/crowdflower/twitter-airline-sentiment.

[19] URL: https://www.kaggle.com/manchunhui/us-election-2020-tweets.

[20] URL: https://pypi.org/project/googletrans/.