Political Opinion Mining using Recurrent Neural Network

Bedanta Das
India Institute of Technology Hyderabad
India
cs17mtech11009@iith.ac.in

Rohan Tondulkar
India Institute of Technology Hyderabad
India
cs17mtech11028@iith.ac.in

Rajat Saini
India Institute of Technology Hyderabad
India
cs17mtech11002@iith.ac.in

Vineeth N Balasubramanian
India Institute of Technology Hyderabad
India
vineethnb@iith.ac.in

Abstract

A Politician's words often reveal their political ideology. Politicians often use Twitter to express their beliefs, stances on current political issues, and reactions concerning national and international events. Indian politicians nowadays very active on Twitter. Thus is becomes an important work to analyze their twitter activity to get to know more about their stance on various issues. In this paper, we build a model to predict the stance of Indian politicians on various issues based on their tweets. The existing work in this area has been done on U.S. politicians. This is a first of it's kind political bias prediction on Indian politicians. Also, as their is no proper labeled dataset for Indian politicians, we build a new dataset to predict issue as well as stance on a new tweet. We do this using a multi-head classification model based on Recurrent Neural Networks (RNNs). Our results show that our model performs similar to existing work but on a relatively smaller dataset.

1 Introduction

The Political space in India is dominated by two major parties Indian National Congress (INC) and Bharatiya Janta Party (BJP). Indian politicians heavily use the Social Networking sites like the Twitter to get connected to general public. Twitter is becoming a digital battleground. These two groups have different political ideologies on various issues like GST, Demonetisation, Inflation, Economic growth, Security etc. Hundreds of politicians tweet and millions of the people re-tweet it. Twitter is becoming a data store for sentiment analysis. Even the political parties are using the data mining techniques during the elections to have maximum benefit from the data. Twitter is the best place to get the tweets from the politicians and mine the opinion of the politicians using the deep learning techniques.

Previous work on sentiment prediction of the politicians is very rare especially in case of Indian political scenario.To the best of our knowledge we are first to use the deep learning technique that too on the Indian politician tweets dataset. With the rise of the Deep Learning in the Natural Language processing, people have started to apply these techniques to detect the private states such as opinions, sentiment, and belief. Models like Recurrent neural network deploy deep architecture with multiple hidden layers and perform well for sequential prediction of tasks.

32nd Conference on Neural Information Processing Systems (NIPS 2018), Montréal, Canada.

2 Related work

Most of the previous work is done on the US Congressional Debate transcripts or the US politicians tweets. Also very few of the previous work use deep learning techniques for sentiment analysis like Actionable and Political Text Classification using Word Embeddings and LSTM [6] uses the LSTM [2] network to perform two types of classification. First is they build models to classify social media messages from customers of service providers as Actionable or Non-Actionable. Second is the classification of social media messages as Democratic or Republican. They showed that LSTM network with word embeddings vastly outperforms traditional techniques. Political Ideology Detection Using Recursive Neural Networks [3] uses the RNN on the Convote dataset (Thomas et al., 2006) which consists of US Congressional Floor Debate Transcripts, Political Bias Analysis [5] uses the LSTM network on the Ideological Books Corpus [1] which is compiled by a group of researchers from the University of Maryland. Modeling of Political Discourse Framing on Twitter [4] presents the problem of issue-independent framing analysis of U.S. politicians on Twitter.

To the best of our knowledge we found no work been done on the Indian Politicians tweets. We created our own dataset of the tweets by the major political parties and the politicians of the country.

3 Methodology

For our problem we have used what we call a Multi-Head classification model as shown in Figure 1. Given a tweet it predicts two labels viz. the issue that is addressed in the tweet and the sentiment of the user. We encode the tweet using an encoder which encodes the tweet into a thought vector which is then fed into the predictor which consists of two classification heads to predict the issue and the sentiment.

For the Encoder we have used GRU, a variant of Recurrent Neural Network to build the thought vector and for the Predictor we have two Fully connected feed-forward Networks (FCFFN) to predict the class labels.

3.1 Gated Recurrent Units

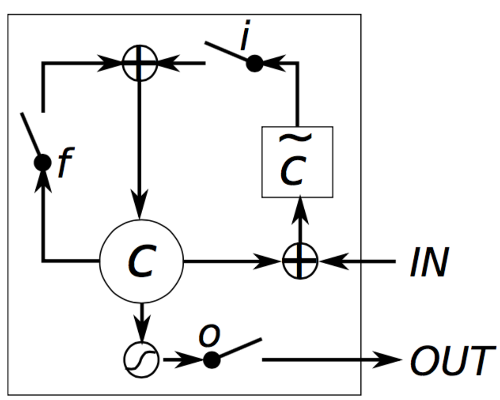


Figure 1: Gru Cell

The GRU[1] is based on Recurrent Neural Network. It retains the ability to resist vanishing gradient problem and it has a simpler internal structure than LSTM[6] which makes the training faster. The GRU cell has two gates, an update gate z, and a reset gate r. The update gate decides how much of the previous memory needs to be remembered and the update gate determines how to combine the current state to the previous memory.

[1]https://people.cs.umass.edu/ miyyer/ibc/index.html

The following equations define the gated mechanism of a GRU :

$$z = \sigma(x_t U_z + s_{t-1} W_z)$$
$$r = \sigma(x_t U_r + s_{t-1} W_r)$$
$$h = \tanh(x_t U_h + (s_{t-1} * r) W_h)$$
$$s_t = (1 - z) * h + z * s_{t-1}$$

3.1.1 Multi-Head Classification Model

The Multi-Head classification model consists of two components : Encoder and Predictor.

The Encoder is implemented GRU unrolled through time and builds a $256 - dim$ vector which is the last hidden state of the Encoder which we call the thought vector. The input to the encoder is a tweet and at each time step a learned embedding of the input word is passed. The number of unrolled time steps is the number of words in the input tweet.

The Predictor consists of two FCFFN both takes the thought vector as input and separately predicts the issue and the sentiment labels. Both networks consist of two layers followed by a softmax layer which gives a distribution over the issues and sentiments.

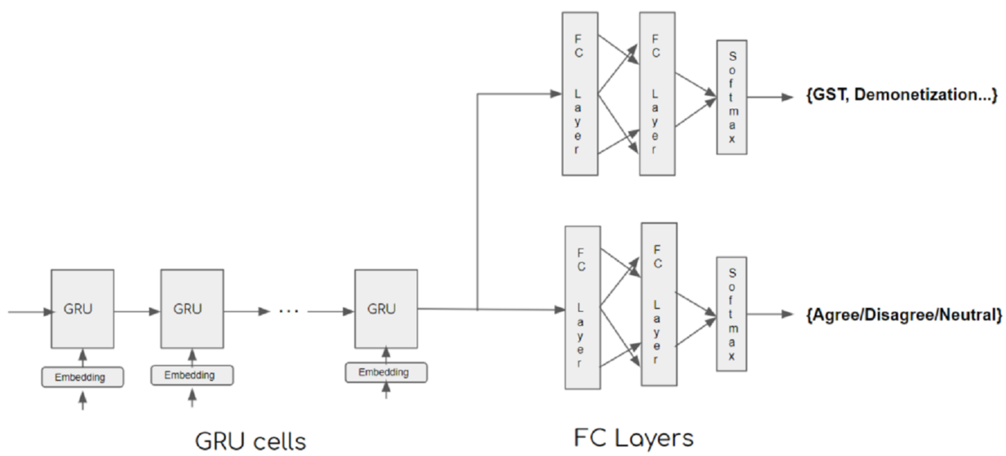


Figure 2: Multi-Head Classification Model

4 Experiments

4.1 Dataset

As mentioned before, all the previous work in this domain U.S. politicians data. Some of the work has been on done using the U.S. Congress debates while the rest has been done on tweets collected for U.S. politicians. To the best of our knowledge, no work has been done in this domain on Indian politicians. As a part of this work, we aim to publish a new dataset in this domain based on Twitter data and tailored specifically on Indian political scenario.

We go about building the dataset in the following way:

List of twitter handles We create a table of twitter handles of 154 Indian politicians along with their party names. It is ensured that politicians from across different parties, national as well regional are taken into consideration. Politicians from 17 political parties in India are mentioned. Along with politicians, we also find twitter handles of party accounts from which a lot of tweets are posted.

Tweet collection We use python tweepy[2] library to collect tweets. Tweets are collected for all the politicians and party accounts. We can collect a maximum of 3200 tweets per twitter account. At the end of this step we have total of 2.8 Lac tweets.

[2]http://www.tweepy.org

Extracting retweet context In case of quoted retweets on Twitter, the original tweet is not returned by the api. Instead the url of original tweet is mentioned. In this case, we crawl the mentioned url to extract the original tweet content and append it to the tweet. We believe this helps the model as we get more information on the original topic which would be regarding a particular issue.

Translating regional tweets As mentioned above, we collect tweets for many regional politicians who tweet in local languages. We cannot miss this information as it is important when working with local issues. We use Google Translation api to translate all the non-English tweets into English.

Text preprocessing We process the textual content of all tweets by considering all urls, user mentions and emojis as separate tokens. Also we remove tweets with less than 4 words in it.

Collecting issues and keywords We dug out various current and trending issues in India. It includes some issues on a national scale like GST, Demonetisation, as well as some local issues like Cauvery dispute, Ram Mandir etc. We also found keywords for every issue based on the tweets posted on those issues. They keywords could be hashtags or non-hashtag keywords. We found 24 issues for this dataset. We can see keywords for few topics in Figure 3.

Issues	Keywords
GST	GST
GST	gstfornewindia
GST	gst4newindia
GST	one nation one tax one market
Demonetisation	Demonetisation
Demonetisation	Demonetization
Demonetisation	Black money
Demonetisation	fake currency
Demonetisation	demonetised
Ram Mandir	ram mandir
Ram Mandir	WeNeedRamMandir
Ram Mandir	babri masjid
Ram Mandir	ram janmabhoomi
Ram Mandir	RamMandir
reservation	SC/ST
reservation	OBC
reservation	backward class
Cauvery SC Verdict	Cauvery
Cauvery SC Verdict	CauveryWaterManagement
Cauvery SC Verdict	Cauvery Management Board
Cauvery SC Verdict	CauveryIssue

Figure 3: Example of keywords per issue.

Assigning stance We make an assumption here that every party has a fixed on a particular issue. There by, all politicians belonging to that party are assigned the same stance as that of their party. We use relevant information and assign a stance as Agreement/Disagreement/Neutral for each political party for every issue.

Labeling dataset We assign labels to each tweet for two things - Issue and Stance. The issue that is assigned to the tweet is the for whom maximum number of keywords are present in the tweet content. The stance is assigned after the issue. It is the stance of the political party on the assigned issue to which the poster of the tweet belongs to. Only the tweets that contain keywords of any of the topics are considered in dataset. After this step, we have 8356 labeled tweets. Figure 4 shows stats on dataset after labeling. In Figure 4a, we see distribution of tweets on issues. In Figure 4b, we see distribution of tweets for all political parties.

4.2 Experimental Details

We split the dataset in to train, validation and test in the ratio of 80:10:10. The number of rolls for the GRU is decided on 60 as we find most tweets have less than 60 words. We only remove selected stop words which would not reduce any information needed to predict Issue/Stance. We use word embeddings to convert word index to vector. The output size of word embeddings is 256. Also, the

Issue	Tweets
Aadhar linking	158
Beef Ban	148
Cauvery SC verdict	183
Demonetisation	1429
EVM tampering	193
FDI Policy	189
Fodder Scam	71
GDP Growth	397
GST	2937
Inflation control	254
Jallikattu Ban	35
PNB Scam	367
Padmavati Screening	111
Ram Mandir	262
Right To Privacy	39

(a) Issue-wise distribution of tweets.

Party	Tweets
AAP	52
AIADMK	181
AIMIM	44
Akali Dal	36
BJP	3721
BJD	7
DMK	139
INC	3177
Janata Dal	192
Janata Party	1
Marxist	164
NCP	150
RJD	81
SP	43
Shiv Sena	81
TMC	271
TDP	16

(b) Party-wise distribution of tweets.

Figure 4: Dataset stats after labeling.

size of hidden vector in GRU is kept at 256. During training phase, we keep the batch size as 32 and train for 5 epochs and each epoch is divided in 5 iterations. We use mini-batch gradient descent with Adam for updating learning rate.

4.3 Results

(a) Loss v/s iterations for Issues.

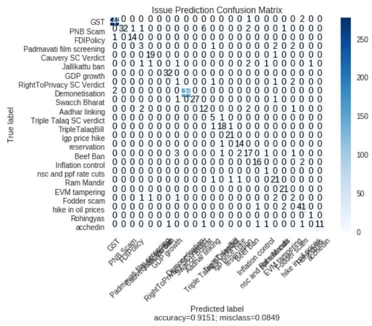
(b) Loss v/s iterations for Stance.

Figure 5: Training and validation loss v/s iterations.

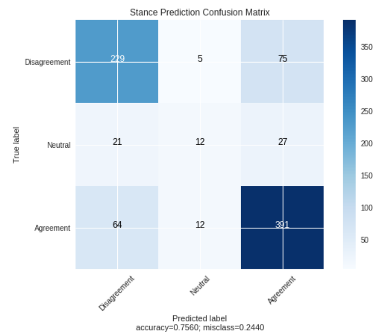
We see the plot of training and validation loss after every iteration in Figure 5. As expected we see that validation loss is higher than the training loss. We see a lot of peaks in training loss but a smoother curve in validation loss. Table 1 shows the results on test dataset. We get 91.51% accuracy for Issue prediction which is a relatively easier problem and 75.6% accuracy on Stance prediction. As it is a multi-class classification, we analyze the confusion matrix in Figure 6. We see in Figure 6b that our model performs well overall but not on Neutral labels. In Figure 5a, we see that on 6 occasions our model miss-classifies PNBScam as Adhar Linking. In rest of the cases, we see there are no more than 1 or two miss-classifications.

Table 1: Results on Test dataset

	Issue Prediction	Stance Prediction
Test Loss	0.334	0.676
Test F1 Score	91.51%	75.6%
Test Accuracy	0.914	0.746



(a) Confusion matrix for Issues.



(b) Confusion matrix for Stance.

Figure 6: Confusion matrix on test dataset.

5 Conclusion and Future Work

In this paper, we build a first of its kind dataset for Stance and Issue prediction based on Indian politicians Twitter activity. We use a multi-head classification model using RNNs to show good classification results on this problem.

In the future we plan to improve our model and incorporate Attention mechanism on top of our model.We are also planning to collect more data to increase our dataset and auto labeling using unsupervised learning.

References

[1] Junyoung Chung, Çaglar Gülçehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling, 2014.

[2] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, November 1997.

[3] Mohit Iyyer, Peter Enns, Jordan Boyd-Graber, and Ps Resnik. Political ideology detection using recursive neural networks, 06 2014.

[4] Kristen Johnson, Di Jin, and Dan Goldwasser. Modeling of political discourse framing on twitter, 2017.

[5] Arkajyoti Misra and Sanjib Basak. Political bias analysis. 2016.

[6] Adithya Rao and Nemanja Spasojevic. Actionable and political text classification using word embeddings and LSTM. *CoRR*, abs/1607.02501, 2016.