

Election Result Prediction Using Deep Learning Techniques

S. JayaKumar, Priyank Patel, Rajat Kumar Singh, Shivraj, Akkshansh Paul

Abstract: A political landscape of any country has always been complex in nature, this complexity can be attributed to several factors such as number of parties, policies and most importantly mixed public sentiment. The rise of social media has given people all around the world to converse and debate with a very large audience, the sheer amount of exposure a tweet or a post received is unprecedented. Recent development in the field of deep learning has led to its usage in several different verticals. Techniques like LSTM allow for performing a sentiment analysis of the posts. This can be used to predict the overall sentiment of the masses in relation to a political party or an individual. Several studies have shown how to approximately predict public opinion, such as in political elections, by analyzing user activities in blogging platforms and on-line social networks. Machine learning has a dramatic growth in recent years and it has been applied in every technology from self-driving cars to e health sectors. We proposed a machine learning models to predict the chances of the winning the upcoming election based on the user or supporter opinions on the social media platform. In Social media the supporter or user express their opinion or feedback post for their favorite's party or opposite party. We have to collect the text posts regarding the election and political campaigns, then we have created the machine learning models to predict the results.

Index Terms: Machine Learning, LSTM (Long short term memory), Data pre-processing.

I. INTRODUCTION

Election is a process carried out in a nation in order to elect a head of state and government officials, and election prediction involves predicting the results for a future elections using data relevant to the said election, in this case tweets. In Existing so many researchers have been done lots of works in the accurate prediction of the upcoming election results. A post-election analysis with the goal of determining where the existing popular prediction models leading up to elections failed to capture the true outcomes of the election.

Manuscript published on 30 April 2019.

* Correspondence Author (s)

S. JayaKumar, Assistant Professor (S.G), Department of Computer Science and Engineering, SRM Institute of Science and Technology, Ramapuram, Chennai, India

Priyank Patel, Student, Department of Computer Science and Engineering, SRM Institute of Science and Technology, Ramapuram, Chennai, India

Rajat Kumar Singh, Student, Department of Computer Science and Engineering, SRM Institute of Science and Technology, Ramapuram, Chennai, India

Shivraj, Student, Department of Computer Science and Engineering, SRM Institute of Science and Technology, Ramapuram, Chennai, India

Akkshansh Paul, Student, Department of Computer Science and Engineering, SRM Institute of Science and Technology, Ramapuram, Chennai, India

© The Authors. Published by Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP). This is an [open access](https://creativecommons.org/licenses/by-nc-nd/4.0/) article under the CC-BY-NC-ND license <http://creativecommons.org/licenses/by-nc-nd/4.0/>

Furthermore, we will construct a model to attempt to predict the election results accurately and efficiently. Thus, prediction based on social media analysis can result in new outcomes, complementary with or even more accurate than traditional prediction poll results. In this paper, a new method for election prediction based on data set analysis is proposed and applied to predict the election results before ahead.

The use of the Social media micro-blogging platform as a tool to predict the outcomes of social phenomena is a recurrent task in the recent social Media. Successful studies can be found in different contexts using various social media for predictive tasks: from prediction of stock market to movie sales, and pandemics detection. A relevant context which received a lot of attention is the prediction of elections and opinions on political events and decisions.

Deep learning algorithm's performance is primarily depended on the data used in training process. It is very critical for a training dataset to be diverse and contain the relevant features in order to obtain a higher accuracy on the real world data. The dataset is to be divided into two parts training and testing in the ratio of 2:1 respectively. First thing is that collect the dataset regarding the political review, that dataset consisting, user opinions, mentions for a given candidate or a political party shows that volumes of mentions of parties reflects the distribution of votes in the election among six parties. Dataset with political related blogs or texts should be pre-processed so, that it will fit under the machine learning model. Data pre-processing, involves Cleaning, Formatting, Sampling, and Normalization, has multiple steps and the number of steps depends on the type of data file, nature of the data, different value types, and more.

II. LITERATURE SURVEY

Election prediction is a very popular problem amongst the machine learning community. In the past various approaches have been taken in order to solve the problem. Random forest and decision tree algorithms are among the popular ones. However, they have their own limitations and often fail to capture essential complex features involved in election prediction. A deep learning model would prove to be more successful in overcoming the limitations of traditional approaches. LSTM has been widely used in NLP tasks. LSTM essentially has a memory like feature which allows it to better capture features, in the said task traditional methods are unyielding. Election prediction task intuitively requires a model to identify and weigh the features, same reasoning is used in natural language classification task where LSTMs are widely popular.

III. PROPOSED SYSTEM

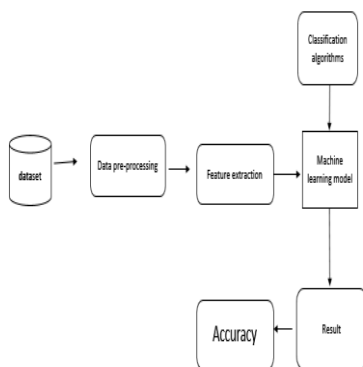
The proposed system consists of a LSTM. The goal was to determine the optimum tuning of hypermeters in order to increase the accuracy of the model.

Advantages

- More accurate result can be possible by applying the LSTM architecture due to its said memory.
- Better Time efficiently.
- Overcome traditional flaws with latest technology and methods.

In this paper, we introduce a model which utilizes LSTM in order to carry out the task of election prediction. LSTMs are a variant of RNN which consists a module in a repeated pattern. The advantages of using the LSTM are defined in section 5.1. The dataset is gathered from twitter using its APIs. The usage of LSTMs allow the model to deal with problems such as vanishing gradient and long term dependency problems.

IV. SYSTEM ARCHITECTURE



The model implements the LSTM architecture in order to extract features from the dataset, LSTM works best for such a diverse dataset with many features. In this section we will discuss LSTM in more depth.

A. Dataset

First, we need to get Twitter API keys. It can be done using 4 pieces of information namely, API keys, API secret, Access token, Access token secret. To get all these 4 elements, first login to twitter account then click on "Create new app". Fill the form and then click on create your twitter application. A next page will be displayed, there click on API keys and copy your API key and API secret. Go down and click create my access token and copy your Access token and Access token secret. Tweepy python library is used for connecting the Twitter API and to get data from Twitter. Python file has been created in which code will be copied. Run the program by using the command: python filename.py then we will see the data from the twitter. Copy the data into a file for analysis purpose.

LSTM

Long Short Term Memory networks (LSTM) are a special type of Recurrent Neural Networks (RNN). They were introduced in a paper by Hochreiter & Schmidhuber (1997), and subsequently defined and popularized by many people. LSTMs properties allow it to be used for a wide variety of tasks, we will now discuss these properties in detail.

LSTMs are mainly preferred because they are designed to

avoid long-term dependency problem. This problem is faced by traditional methods as they struggle to remember information over a long periods of time, however, LSTMs do not struggle in this area.

All RNNs are inherently repeating in nature, and LSTMs are not any different. However, there is one important difference in the repeating module; this difference is what separates LSTM from RNN. LSTMs have four neural network layer in a single module, they interact in a very special manner.

A LSTM module has four gates (layers), they are:

- Forget gate layer
- Input gate layer
- Update gate layer
- Output gate layer

Here, o , f are called the input, forget and output gates, respectively. Note that they have the exact same equations, just with different parameter matrices (W is the recurrent connection at the previous hidden layer and current hidden layer, U is the weight matrix connecting the inputs to the current hidden layer).

1) Forget gate layer

The forget gate layer is kept in the beginning of the module. It is the most distinctive layer in LSTM. The forget gate layer is tasked to decide which information is of no importance, and thus can be 'forgotten'. This layer allows the LSTM to perform very well over a longer period of time; an essential quality which helps in the learning task. The layer achieves its task by using a sigmoid layer.

2) Input gate layer

The next step is to decide what information we are going to store in the state. First a sigmoid layer decides which information is important and needs to be stored. In the next step a tanh layer is used to create a vector of new values, which would be added to the state.

3) Update gate layer

The update gate layer is used to update the old state into a new state, which is intuitively the immediate state. Here we take a product of the old state with the vector of the forget gate layer, this allows us to discard the information we decided to forget.

4) Output gate layer

The output gate layer is the final state in a LSTM cell state. It is used to decide which values we ought to output. The first step involves using a sigmoid layer, this said layer decides which information we are going to output. Next step involves the use of tanh function to set the range of values between -1 and 1, now we take the product with the sigmoid layer; thus selecting only the required values.

V. MODULE IDENTIFICATION

This section handles the prominent modules that are being used in this model which goes sequentially to carry the assigned task. These modules are developed after certain studies for the models that were designed earlier but created some issues with the overall network.



The modules are as follows;

- Data preprocessing module
- Training module
- Testing module

A. Data Pre-processing Module

In this module the data is being collected by using the twitter API. Then the data is stored in a tabled form, which is ideal for classification tasks. Now the data is randomized in order to remove any pattern, if any.



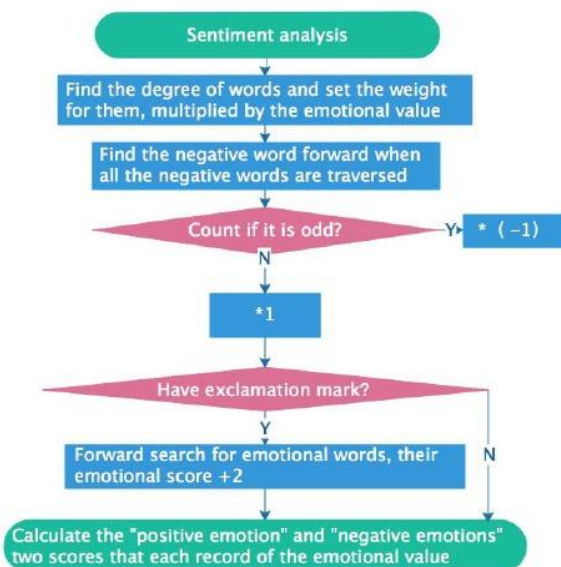
Next step involves dividing the data into training and testing parts. The training part will consist of approximately two-third of the whole dataset and the remaining dataset will be allocated for the testing phase.

B. Training Module

The training module requires a feature extraction to be done before hand. This will allow the algorithm to identify the required features, which are required to identify the object, in this case the sentiment of the post. Now comes the training part, for training we use random forest algorithm as it proven to give best results in datasets such as ours. The training phase will use the training part of the dataset and not the testing part. The result will be saved in a model, which is achieved by NLTK library of python. The training phase will be done for 10000 epochs.

C. Testing Module

The testing module uses the model saved by the training module. The dataset used in this module is the remaining one-third of the original dataset. In this module our model's results are compared with the expected output. Subsequently, the accuracy of the model is calculated, the said accuracy is considered to be the ideal representation of how our model will work in the real world.



VI. RESULT

The model performs very well giving accuracy of 85%. The model was trained on data on Indian elections retrieved from Twitter. Tracking Indian politic stage can prove to be very complex given the number of political parties, population and the language variations. However, the model still gives a commendable accuracy of 85%.

VII. DISCUSSION

The decision to use a LSTM was an intuitive one which resulted in such an accurate result. LSTM has been used in natural language classification tasks and due to its memory feature it has given state of the art results in the NLP space. Reasoning behind this is that LSTM can successfully capture the important features and thus when a model can track such features and thus make decisions with more accuracy. Similarly, LSTM here also gives more accurate results than traditional models and the variance of the decision across different elections is reduced by a great measure, which in turn translates into a more robust model.

VIII. FUTURE ENHANCEMENTS

There could be many other prospective areas to conduct this research in, including the data from other big social media sites like Facebook to increase the variance of the data set. We have more space to work with the training dataset such as considering the sample dataset in which the certain number of features of an algorithm is already defined. The number of hidden layers can also be increased and the result can be monitored in order to make note of any pattern. Different activation functions can be used in order to determine which works best with the dataset. The hyperparameters can be tuned to obtain a further increase in the accuracy by a small magnitude; it would not be able to obtain a great difference in the accuracy of the model, but one can learn more about the workings of the model by performing hyperparameters tuning.

IX. CONCLUSION

In this paper, we take on the task to predict the Indian elections by performing sentiment analysis of the twitter data. The dataset consists of the real tweets. The problems encountered during the course of this project was to collect a meaningful and large dataset, enough to obtain a high accuracy. The model uses a LSTM architecture in order to better focus on important features and thus the model is said to be making a more informed decision. The primary reason for using a LSTM architecture was to overcome the vanishing gradient problem. Vanishing gradient is a condition in which a model tends to focus on a limited set of features after a certain training time and as a result the model cannot perform as intended in the real world problems. Another problem linked with the task was the diverse nature of Indian political space (Most of the data was collected in this space).



It is quite difficult for traditional machine learning algorithms to capture the essential features required to produce high accuracy. The LST model overcomes the limitations of traditional deep learning and machine learning techniques. Our model achieves an accuracy of 85% which is very high for the task at hand.

REFERENCES

1. Lei Wang and John Q Gan ,“Prediction of the 2017 French Election Based on Twitter Data Analysis”.
2. Pritee Salunkhe, Sachin Deshmukh, “Twitter Based Election Prediction and Analysis”.
3. Varghese S Chooralil, Rincy Jose, “Prediction of Election Result by Enhanced Sentiment Analysis on Twitter Data using Classifier Ensemble Approach”.
4. Dongdong Guo, Meng Li , Anny Bian, “The Prediction on the Election of Representatives”.
5. Pritee Salunkhe, Avinash Surnar, Sunil Sonawane, “A Review: Prediction of Election Using Twitter Sentiment Analysis”.

AUTHORS PROFILE



S. JayaKumar, Assistant Professor (S.G), Department of Computer Science and Engineering, SRM Institute of Science and Technology, Ramapuram, Chennai, India



Priyank Patel Student, Department of Computer Science and Engineering, SRM Institute of Science and Technology, Ramapuram, Chennai.



Rajat Kumar Singh Student, Department of Computer Science and Engineering, SRM Institute of Science and Technology, Ramapuram, Chennai.



Shivraj Student, Department of Computer Science and Engineering, SRM Institute of Science and Technology, Ramapuram, Chennai.



Akkshansh Paul Student, Department of Computer Science and Engineering, SRM Institute of Science and Technology, Ramapuram, Chennai.