State-wise sentiment analysis of NRC, CAA and NPR tweets

Harshita Pandey
MT19012
harshita19012@iiitd.ac.in

Shivani Mittal
MT19128
shivani19128@iiitd.ac.in

Rupali
MT19095
rupali19095@iiitd.ac.in

Abstract

People's Opinions and views help us analyze how the propagation of information impacts the lives of a large number of people. The twitter data is being increasingly used to determine consumer's sentiment towards a scheme or product. In this project, we have used the twitter data to study the impact of CAA, NRC and NPR state-wise across the nation. The goal of this task is to discover the opinion of the tweets, which is typically formulated as a machine learning based text classification problem.

1 Introduction

Social networks are the main resources to gather information about people's opinions and sentiments towards different topics. Twitter, being the one such popular social website contains an abundance of such data that can be processed in order to obtain meaningful results like sentiment scores, product reviews, and predictive analysis. The main objective of this project is to use the Twitter data to study the nation-wide opinion and views on CAA, NRC, and NPR using sentiment analysis. Also termed as opinion mining, sentiment analysis is primarily used for analyzing the opinions and conversations of the public and using this data to classify the sentiment as positive, negative or neutral.

2 Data Extraction

Twitter is a platform where the public shares its opinions. Twitter provides us with the "tweepy" library for accessing the Twitter API. To authenticate the twitter API, four keys are required: Consumer Key, Consumer Secret, Access key and Access Secret. The followings steps are required to access these keys:

1. Create a developer account on the twitter platform.
2. Create an application and fill the required details.
3. After creating the application, details of the app are shown along with the Consumer Key and the Consumer Secret.
4. The Access Token and the Access Secret Token are generated in the "keys and token" section.

Now, authenticate the twitter API using the generated keys. Extract the twitter data using the tweepy library. The various fields extracted in the data are: Location, Retweet Count, Date and Tweets.

3 Data Pre-processing

1. **Remove duplicate rows:** In the extracted data, duplicates tweets can be present. The two main reasons for this can be : Tweet-collector collects the same tweet multiple times or different users may post the same tweets[1].
2. **Removing References:** Most of the tweets are in reply to another user which uses @username for the reference. There is no use of this referenced user for the sentiment score, and hence we have removed it from the data for the same.
3. **Special Character:** The data contains some special characters. A dictionary is maintained which contains all special characters along with their meanings. All the special characters in the tweets are replaced by their meanings.
4. **Replacing Abbreviations:** A separate file is maintained in which abbreviations and their full forms are present. In the data, these abbreviations are replaced by their full forms.

5. **Spelling correction:** In the data, the incorrect spellings are corrected using the "correct function" in the "TextBlob" library.

6. **Word Normalization:** There are two techniques to normalize a word: Lemmatization and Stemming. Lemmatization converts the tokens to its base form or a valid dictionary word whereas stemming converts the tokens to their root forms. Suppose there is a token 'saw,' stemming might return just 's' but lemmatization would return either 'see' or 'saw' depending on whether the use of token was as a verb or a noun. We have used lemmatization for word normalization.

7. **Language Translate:** There are some tweets in the data that are in other languages such as French or Hindi. To get the sentiment for the text present in these different languages, all the text has been converted to English.

8. **Locations:** The data is stored in the form of a dictionary, where the keys contain the states and the values store the tweets corresponding to those states. There are some tweets that contain the location in the form of cities and thus are replaced with their respective states. The tweets which do not contain any location or contain special characters are stored as random tweets.

4 Methodology

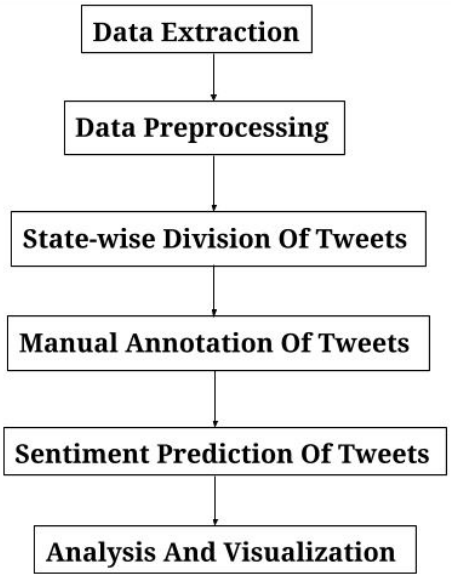

Figure 1: Architecture

4.1 Sentiment Analysis

It is the process of computationally identifying and categorizing opinions expressed in a piece of text, especially in order to determine whether the writer's attitude towards a particular topic, product, etc. is positive, negative, or neutral. We manually labelled the sentiments of tweets. We labelled -1 if the user's opinion is against the implementation of CAA, NRC, NPR, and +1 if the user is in the favor. We also assigned a separate category for the users which are neither against nor in the favor. These users may be against the violence or criticized the media for hate mongering among the public.
We annotated 4062 tweets and used it for building the model and also analyzed this data with the help of graphs. The figure 2 shows the count of postive, neigative and neutral views of the public. From the graph it is observed that most of the people are neither in favor nor against of the NRC, CAA and NPR.

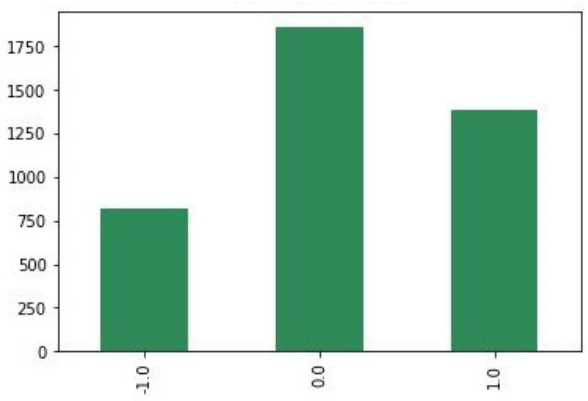

Figure 2: Sentiment Distribution

4.2 Encoding the words

We cannot directly feed the text data into the machine learning models. So, before training the data, we need to encode the text into number. We used two techniques to numerically represent the data:

1. **Text to sequence:** First, tokenize the text using the tokenizer class. Afterwards, take each word in tweet and replaces it with its corresponding integer value from the "word_index" dictionary. It will represent the sequence of text into variable length number sequence. But, to feed the data into machine learning model, the input sequence must have the same length. So, to deal with the problem of short and long tweets, we pad the sequence with 0

using the "pad_sequences" method available in the "Keras" library.

2. **Tf-idf vectorizer:** Tf-idf vectorizer converts a collection of raw documents into a matrix of tf-idf features. The tf-idf score is the product the tf value and idf value for each term.

    - **Term Frequency:** Term frequency of term t is the number of times term 't' present in the document.
    - **Inverse Document Frequency:** Document frequency is defined as the number of document 'd' that contain the term 't'. Inverse document frequency is the reverse of document frequency.

3. **Count Vectorizer:** Count Vectorizer converts the collection of text documents to a vector of token counts. Basically, it represents each token with its term frequency.

4.3 Machine Learning Models

In this section we build machine learning models to predict the sentiments of the tweets. This prediction will help us to analyze the views of the public on the implementation of CAA, NRC and NPR.

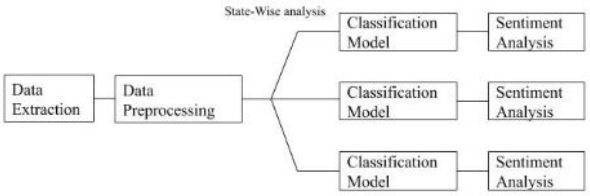

Figure 3: Methodology

4.3.1 Random Forests

The Random Forest consists of a large number of decision trees, which work together as an Ensemble model. It is based on the concept that the uncorrelated models when combined together may produce more accurate results as compared to the individual predictions. The steps of the Random Forest are:

1. It begins with the original dataset as the root node.

2. For each attribute, it find the entropy or information gain and selects the attribute having the smallest value.

3. The selected attribute acts as the split node and the remaining dataset is split based on the condition.

4. The process stops when either the tree reaches the maximum depth or the remaining data points reach a threshold value.

5. The predicted value is the average value of the predictions made by various decision trees.

4.3.2 Support Vector Machines

Support Vector Machines is a supervised algorithm which is used for finding a hyper-plane which clearly classifies the data points of the two classes plotted in a n-dimensional space.
In case of data which is not linearly separable, it makes use of the **kernel Trick**. It takes the data in lower dimensional input space and transforms it to a high dimensional space, thus converting the non-separable data to a linearly separable data.

4.3.3 Linear Regression

Linear Regression is supervised Machine Learning Algorithm, which tries to predict the value of the independent variable, given a set of dependent variables. The different Regression models differ on the type of relationship that exist between the variables and the number of independent variables. The algorithm basically tries to find out a linear relationship between the input and output variables and hence the name linear Regression.

4.3.4 Stochastic Gradient Descent

Gradient Descent Algorithm can be defined as an optimization algorithm, which tries to find the values of the parameter of the function, which minimizes the cost function. In case of stochastic gradient descent, rather than taking the whole dataset together, it considers only a sample data point. Though it may be a noisy approach but efficient as long as we reach the minima in a very small training time.

4.3.5 Recurrent Neural Network

In neural networks, the input and output are generally independent of each other, however in case of Recurrent Neural Network, the output of the previous step is used as the input for the next step.
RNN is generally used with sentiment analysis, which create a neural network, models it and predicts the probability of each class.

Models	Text to sequence	Tf-idf vectorizer	Count Vectorizer
Logistic Regression Model	45.75	61.5	59.18
Stochastic Gradient Descent	30.67	50.67	60.12
Support Vector Machine	52.02	62.97	67
Random Forest	55.55	58.056	59.40
Recurrent Neural Network	59.02	61.6	50.01

Figure 4: Accuracy on machine learning models using different encoding techniques

Now, to check the performance of the models, we manually verified the predictions on the test dataset. It was observed that SVM model performed better than any other classification algorithm. From the table 4 it is also observed that model trained on the sequence of text represented using tf-idf score and term frequency are performing better than the text to sequence representation.

5 Challenges faced in Sentiment Analysis

- The training and testing dataset is not large enough to train models efficiently. The models are trained on the manually annotated data and it is not practically possible for us to build a large annotated dataset.
- In the tweets, most of the users did not show their viewpoint on NRC, CAA implementation. Instead of it, they criticized media and political leaders for hate mongering among the people. This made difficult for us to understand their views on NRC, CAA.
- The tone of some tweets was sarcastic, and it is a challenging task for model to accurately predict the sentiment of the user.

6 Visualizations

Library Used: Geopandas, Flask
Geopandas adds a geometry column to the DataFrame. This geometry column contains Shapely geometries (Shapely is a Python library with geometry types and operations), which means that we can access all the properties and methods available to Shapely objects directly on the DataFrame, or even the feature itself. For accessing geometry "India_States.shp" file is used which returns a dataframe as in Geometric Dataframe in Figure 3.

	st_nm	geometry
0	Andaman & Nicobar Island	MULTIPOLYGON (((93.71976 7.20707, 93.71909 7.2...
1	Arunanchal Pradesh	POLYGON ((96.16261 29.38078, 96.16860 29.37432...
2	Assam	MULTIPOLYGON (((89.74323 26.30362, 89.74290 26...
3	Bihar	MULTIPOLYGON (((84.50720 24.26323, 84.50355 24...
4	Chandigarh	POLYGON ((76.84147 30.75996, 76.83599 30.73623...

Figure 5: Geometric Dataframe of States of India

Now, this needs to be merged with dataframe containing the percentage of positive, negative, and neutral tweets corresponding to each state. Now we can plot this merged dataframe into India map based on the attribute we want. Here is the final map obtained after plotting wrt positive and negative columns.

Front End: We tried to implement the front end using Flask. Flask's design is lightweight and modular. Therefore, it is easy to transform it into the web applications or framework when one needs very few extensions without weighing much. We used html5 and CSS for providing an interactive user interface at the back of the python file. Flask acts as an intermediate for HTML and python.

6.1 Analysis with Map:

Based on Negative Tweets:

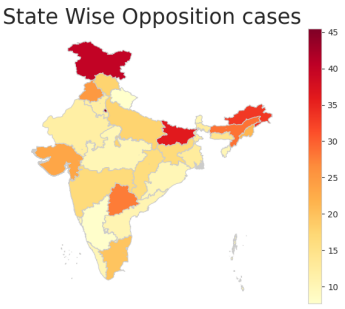

Figure 6: Map Representing States in Opposition

As shown in Figure 4 we have observed that states including Delhi,Jammu Kashmir, and Assam are against the NRC-CAA act this can be due to violence happened in these regions.

Based on Positive Tweets:

As shown in Figure 5 we have observed that in states like Gujrat and Tamil Naidu people are in favour of CAA are in favour of the NRC-CAA act that is why the decision of implementation still

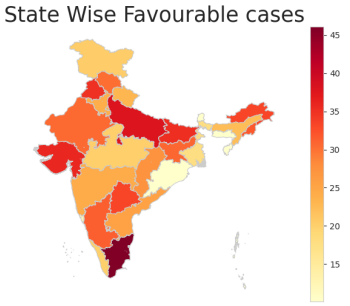

Figure 7: Map Representing States in Favour

holds in these states but in states including Delhi, Kerala, Madhya Pradesh more than 50% people were against this that is why these states may overrule the NRC-CAA act.

References

[1] Symeon Symeonidis, Dimitrios Effrosynidis, and Avi Arampatzis. "A comparative evaluation of pre-processing techniques and their interactions for twitter sentiment analysis". In: *Expert Systems with Applications* 110 (2018), pp. 298–310. ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2018.06.022. URL: http://www.sciencedirect.com/science/article/pii/S0957417418303683.