Sentiment Analysis using Lexicon based Approach

Munna Pandey[1], Rebecca Williams[2], Nikita Jindal[3], Anurag Batra[4]
Assistant Professor, Institute of Information Technology & Management, Janakpuri, New Delhi, India
Research Scholar, Institute of Information Technology & Management, Janakpuri, New Delhi, India
pandey.id@gmail.com, rebecca3williams@gmail.com, nikitajindal68@gmail.com, anuragbatra1999@gmail.com

Abstract - Triple talaq is also known as talaq-e-biddat instant divorce. It is a kind of Islamic divorce used by Muslims in India. It allows Muslims man to divorce their wife legally by simply stating the word 'Talaq' three times in any form which can be in any way (verbal, written, or in electronic form). Now a day, the huge amount of data is posted on daily basis on the social media platform. Twitter is a well known social networking platform where the user can post their views, opinions, and thoughts freely. The sentimental analysis is a process of understanding opinions, thoughts and feelings of people about a given subject. This paper analyses tweets posted on Twitter on the subject Triple from the year 2002 to the year 2019. We have transformed unstructured data into well-informed data for getting the insights of people. The main focus of the work is to analyze the feelings of people using two well-known API like TextBlob, and SpaCy. These APIs are based on Lexicon approach. This paper predicts sentiment into three classes positive, negative and neutral.

Keywords: - Talaq, ApplicationProgramming Interface(API),SpaCy,TextBlob,Parts-of-Speech (POS),Natural Language Processing(NLP)

I. INTRODUCTION

Triple talaq also named as talaq-e-biddat instant divorce. It is an Islamic divorce in the Muslim religion. It allows Muslims man to divorce their wife legally by simply stating the word 'Talaq' three times in any form which can be in any way (verbal, written, or in electronic form).22nd August 2017 was the date when the Indian supreme court instantly deemed the triple talaq. Out of five judges, three judges have the same opinion that triple talaq is illegal and the remaining two stated that the government should ban this practice by simply following the law. The Modi government made a bill known as the Muslim women bill, 2017 and it was passed on 28th December 2017 by the Lok Sabha. The bill stated that if a man gives instant triple talaq in any form- spoken or written or by any electronic means like by email, message or any other mean of communication will be considered illegal and if any such practice found then there will be a three years imprisonment and will be fined. In Islam marriage is considered as a contract between husband and wife and in that various procedures have been written on how to annul it. As per Islamic traditions, a woman can ask divorce via "khula", whereas the husband can end the marriage instantly by pronouncing talaq thrice. But many have highlighted the misuse of instant divorce by men as a reason to ban it. The man will pay some maintenance and custody of the child to the mother. The bill passed should not be viewed from point of politics. The bill passed should not also be viewed from the point of religious motive or for vote bank. The bill is passed for the rights and respect of women. The sentimental analysis is a process of analyzing views of people about a given subject or a topic which can be in the form of written or spoken language. Today in this world where a huge amount of data generated every day, sentimental analysis has become a vital tool for making sense of each processed data. This has allowed companies to get the results of various processes they are doing nowadays. It is a type of text research or mining. In this paper, we are applying statistics, natural language processing (NLP), and machine learning to identify, analyze and extract some important information from tweets. The main objective is to observe the reviewer's feelings, expressions, thoughts or judgments about a particular topic, events, or a company. This type of analysis is also known as opinion mining which has its main focus on extraction. The goal of this analysis is to get the opinion of our audience on a particular subject by analyzing a large amount of data from heterogeneous sources. Today sentimental analysis has a different number of uses. With the increase in the use of social

networking sites and the rise in feedbacks forms and ranking sites, companies are becoming more interested in this type of analysis. Consumers can freely share their thoughts easily on the web. But with a huge amount of information out there, it can be tedious for companies to hone in on the most valuable parts of consumers comment.

This is the main reason why we are applying sentimental analysis. Organizations are leaning on the basis of sentiment analysis and then filtering out valuable information so that they successfully understand consumers behavior on a topic so that they can take targeted decisions. Now a day, the huge amount of data is posted on daily basis on the social media platform. Twitter is a well known social networking platform where the user can post their views, opinions, and thoughts freely.

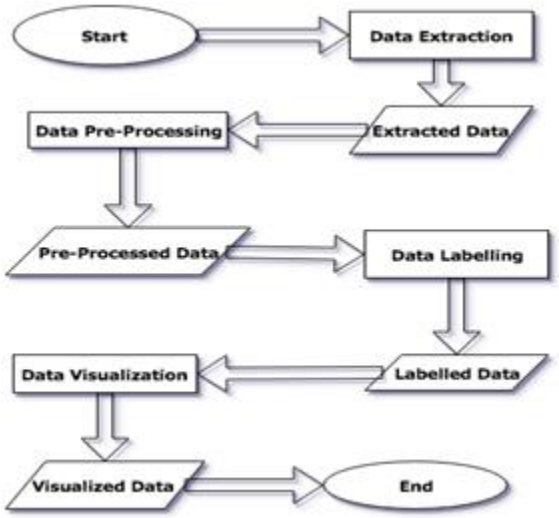

Fig 1. Flow chart of the process

Sentiment Analysis can be done by either machine learning or lexicon-based approach. In this paper, we have applied a Lexicon based approach. This is a feasible and practical approach which can analyze tweet text without training or using machine learning. Lexicon is a collection of words or one can say it is like a dictionary in which words are arranged alphabetically. This approach is subdivided into a dictionary-based approach and corpus-based approach. Here we are using a corpus-based approach. Corpus is a large body of words or text which formulate a set of conceptual rules that govern a natural language from texts in that language and examine how that language relates to other languages.

II. RELATED WORK

In the last few decade there has been the vast expansion of social media due to which social media has started playing a vital role when it comes to gathering sentiments of masses. Something similar was done there are various ways to perform sentiment analysis using lexicon based approach and some of the methods are discussed by Taboada, M., Brooke, J., Tofiloski, M., Voll, K., & Stede, M.in their research paper Lexicon-based methods for sentiment analysis.

III. EXPERIMENTAL SET-UP

The experimental setup of the approach presents the research methodology employed, the tools, and libraries used to analyze the opinion of people of India on Triple Talaq. We used a laptop of DELL, i3, 3.6 GHz with 8GB DDR3 RAM. In this study, open source libraries, packages, APIs are extensively used.

This section is further categorized into two sub-sections.

A. System Architecture

In this section, we are discussing the overall architecture of a system. The tweets that were extracted do not contain their corresponding labels. So, labeling of tweets was required in order to get the desired results. In this paper, we used a lexicon-based approach to classify our tweets. Currently, many web services are available which automatically provide more clear-cut labels as compared to human annotators. The main focus of the work is to analyze the feelings of people using four well-known API like Text Blob, NLTK, SentiWords, and Vivekn. The services provided by these four approaches vary from one another.

a. Data Extraction

With an increase in the importance of text analysis in many research areas, so many researchers started analyzing the sentiments of people by their posts on many social sites. In this paper, we are analyzing the tweets posted by people on triple talaq. The first of analyzing is data extraction. Data extraction is the process of retrieving the data posted by people from

available sites (in this research we are taking data from twitter). After this we preprocessed our data i.e., we cleaned our data (removed noisy data).

b. Labeling of Tweets

In this part, we have labeled our tweets using four well known APIs. The four well-known API that we have used over here are TextBlob, SentiWord, NLTK, and Vivekn. These four APIs label our tweets into three well-known categories (Positive, Neutral and Negative).

c. Data Visualization

In this section, we discussed how after applying all our techniques we Visualize all our gathered information.

- **Pie charts of our results of APIs**

In this, we have visualized how many tweets are classified into which category(positive, negative or neutral) in the form of pie charts for each API.

- **Comparison of our APIs used**

In this, we have compared all our four API and have determined the accuracy of each API that we have used. This accuracy has been shown using a histogram.

B. Tools Used

In this section, we are discussing the various tools that we have during this research. Here we are discussing programming languages, APIs, open source libraries in brief.

- **Tweepy**

Tweepy is open source which enables python to communicate with Twitter and enable us to use its API.

- **Python programming language**

Python is a popular programming language which nowadays is being used for text mining and analysis. It is object-oriented and high-level programming language.

- **Text Blob**

It is a python library for textual data. Using text blob we can tokenize our paragraph into sentences or words.

- **SpaCy**

It is a free and open source library which is used for advanced NLP (natural language processing).

- **Matplotlib**

It is a python plotting library (in 2D). It produces quality figures in a variety of formats. It also helps in plotting various graphs for data visualization.

- **Seaborn**

It is a python library for making statistical graphs.

IV. Data Extraction

A. Extraction of Data

It is one of the most important step cause from here only things start. Now Twitter is the most famous social network where people from all around the world share their views which are also called tweets. Twitter helps us to access its API through python library called Tweepy which allows us to extract the data from the twitter of any user. Tweepy provides an easy to use Cursor interface by which we can iterate through different types of objects. This python library can be installed by using pip command on cmd or terminal.

The first thing we need to know are the keys, that are: the consumer key, the consumer secret key, the access key and the access secret key from Twitter's developer site available easily for each user. These keys will help the API for authentication.

Steps to obtain these keys are as follows :

- Log in to Twitter's developer site.
- Go to “Create an App”.
- Fill the details in the form.
- Then Click on “Create your Twitter Application”.
- Details of our new application will be shown along with consumer key and consumer secret key.
- To obtain access token key, click on ”Create my access token”.

The page will refresh and generate an access token. For authorization of our twitter account, we use

OAuth Interface. By this, we can authorize our app to access our account.

Now, Twitter limits the tweets to be extracted to 3200 maximum at a time. A JSON file is created when we fetch tweets from Twitter using tweeps. Since JSON file data is complex and it contains a lot of unrequired data, we created a corpus file, which contains only required data which can be analyzed.

We also converted time to the timestamp to fetch data between the year 2002 to 2019. Timestamp data is easily read by machine. We have Created separate files for name and text to improve the performance of the code.

Code snippet for extracting tweets

```
import tweepy
import csv
import pandas as pd

def TWITTER(Save):
        #input your credentials here
        CONSUMER_KEY = input("Enter Consumer Key: ")
        C0NSUMER_SECRET) = input("Enter Consumer Secret Key: ")
        ACCESS_TOKEN= input("Enter Access Token: ")
        ACCESS_TOKEN_SECRET = input("Enter Secret Access Token: ")

        AUTH = tweepy.OAuthHandler(CONSUMER_KEY, C0NSUMER_SECRET)
        AUTH.set_access_token(ACCESS_TOKEN, ACCESS_TOKEN_SECRET)
        API = tweepy.API(AUTH,wait_on_rate_limit=True)
        file = STANDARDIZED_CSV(Save)
        csvFile = open(file, 'a')
        #Using csv Writer
        temp0 = get_input()
```

Code snippet to tokenize content

```
import nltk
from nltk.tokenize import word_tokenize
m = nltk.word_tokenize(tex[1].strip().lower()
        for temp1 in temp0:
                csvWriter = csv.writer(csvFile)
                for tweet in tweepy.Cursor(api.search,q=temp1,count=10, lang="en", since="2001-04-03").items():
                        print (tweet.created_at, tweet.text)

                csvWriter.writerow([tweet.created_at, tweet.text.encode('utf-8')])
```

B. Pre-Processing of Data

This is another important step to proceed further is preprocessing. Preprocessing of data is required to clean the data to acquire the required data. In this step, all the noisy characters are removed from the text to further analyze it. The misspelled words, grammatical errors, punctuation errors, unnecessary capitalization, stop words and use of non-dictionary words such as abbreviations or acronyms of common terms are few examples of noise in the text.

A classic tweet contains variations of words, emoticons, mentions of users, hash tags etc. The main goal of the preprocessing step is to standardize the text into a relevant form to derive the sentiments of the user. Following are the steps to pre-process text into useful data for classification:

a. Tokenization-

First of all the text is tokenized. Tokenization is the process of natural language processing(NLP) by which large textual data is divided into smaller parts called tokens. In other words, tokenization helps to subdivide sentences into a group of words and paragraphs in a group of phrases. This step is a crucial step in NLP.

Tokenization can be of two types:

i) word tokenization
ii) sentence tokenization

We are using nltk word_tokenizer() to tokenize our textual data split a sentence into words. Then the output of the tokenization is converted into a data frame.

Now, tokenization of text from the corpus is done in three ways using nltk i.e. unigram, bigram, and n-gram. These text models can also be used in tokenized sentences.

b. POS Tagging-

The second step of pre-processing the data is POS Tagging. Parts-of-speech Tagger is very useful as it reads the text and assigns parts of speech or tokens (i.e. noun, pronoun, verb, adjective, etc.) to each word.

Code example:

```
text = word_tokenize("This sentence is written to check pos tagging in nltk")
nltk.pos_tag(text)
```

Output: [('This', 'DT'),('sentence', 'NN'),('is', 'VBZ'),('written', 'VBN'),('to', 'TO'),('check', 'VB'),('pos', 'NN'),('tagging', 'VBG'), ('in', 'IN'),('nltk', 'NN')]

c. Lemmatization:

The third step of preprocessing is Lemmatization. It is an algorithmic process of finding the lemma of a word depending on its meaning. Lemmatization usually refers to the linguistic analysis of words. The main aim of this process is to remove any inflection in the ending of a word.

Now in text pre-processing both stemming as well as lemmatization. They both seem similar but are different because stemming method cuts the suffix from the word i.e. either the ending or the beginning of a word which sometimes makes the word meaningless. For example: Stemming for studies is studi , which indeed have no meaning in the dictionary.

On the other hand, lemmatization is a much better method and is more powerful as it also considers the morphological analysis of a word which helps in conversion of the word into its base form without changing its meaning. For example Lemma for studies is study.

So we can say that lemmatization is a smart method and stemming is a generic method. Hence, lemmatization will help in creating better machine learning characteristics.

Code Example:

```
from nltk.stem import WordNetLemmatizer
wordnet_lemmatizer = WordNetLemmatizer()
wordnet_lemmatizer.lemmatize("teaches")
```

Output: 'teach'

Also, for verbs, we use “v” as an argument to pos as the default pos argument is “n”.

Code Example:

```
wordnet_lemmatizer.lemmatize("is", pos="v")
```

Output: 'be'

d. Stop words removal

Stop words removal is one of the major pre-processing steps as it is used to filter out useless data.

In natural language, stopwords are the frequently used words such as is, am, are, an, the etc which have very little meaning. These words are ignored by the search engine while indexing entries for searching and retrieving them. The programming languages are programmed to ignore such words. We are removing these words as they are considered as they do not add any value to our analysis.

Code snippet for removing stop words and punctuation in the tweet data

```
import math
from nltk.corpus import stopwords
self = ["`","``",""""]
stop_words = set(stopwords.words("english") + list(string.punctuation) + list(self))
```

e. Translation and Language Detection

Last but not least this step is used to detect and translate a given language into a required language. We are using Textblob which is a python library for this task. Now, textblob is an amazing tool which makes NLP faster and easier to work with and this translation feature of textblob is one of the best features. Let us do some code examples and see how it works:

Code Example:

```
from textblob import TextBlob
blob= TextBlob("مرحبا بالعالم")
blob.detect_language()
```

Output: ‘ar’

Code Example:

```
blob.translate(from_lang="ar", to='en')
```

Output: TextBlob("Hello World")

V. Labeling of Data

This step is further subdivided into four steps including various APIs to label the text into negative, positive or neutral. The APIs we are using are - TextBlob, Nltk, SentiWord, and Vivekn. We are using these four APIS to label the tweet text and then we will compare and contrast them. These APIs are freely available and are usually used in the lexicon-based approach. The pre-processing steps are almost common in all of these APIs i.e. tokenization, POS tagging, stemming and lemmatization, stop words removal etc. Vivekn labels sentiment using words, n-grams, and phrases whereas TextBlob uses Parts-of-Speech (POS) tagging to label the sentiments. Four python scripts are written for labeling the sentiment by accessing their APIs.

A. TextBlob

TextBlob is a python library and it helps in various natural language processing tasks providing a simple API.

Textblob has easy to use interface hence it is beginner friendly. It is fairly simple to learn Textblob when compared to other open source libraries which is one of its key feature.

It is basically used for text processing which includes tasks such as tokenization, Parts-of-Speech tagging, stemming, lemmatization, stopwords removal, translation, noun phrase extraction, sentiment analysis by labeling, classification by using machine learning algorithms, and more. It has a sentiment property which returns a tuple of the form Sentiment (polarity, subjectivity). The polarity score is a floating point number within the range [-1.0, 1.0](where -1 means negative statement and 1 means positive statement). The subjectivity is a floating point number within the range [0.0, 1.0] (where 0.0 is very objective and 1.0 is very subjective). The subjectivity is basically the person's opinion or emotion on a particular topic. If you want to work on basic NLP tasks, TextBlob is the best open source software. In fact, TextBlob performs better than NLTK for textual analysis.

Code snippet:

```
for m in tweets: tem = 3
analysis = TextBlob(m)
if analysis.sentiment.polarity > 0: tem = 2
text_pos = text_pos + 1
elif analysis.sentiment.polarity == 0: tem = 0
text_confused = text_confused + 1 elif
        analysis.sentiment.polarity < 0:
tem = 1
text_neg = text_neg + 1
```

B. SpaCy

It is a free and open source library which is used for advanced NLP (natural language processing). Spacy is gaining popularity at a very fast rate and is said to overtake NLTK. SpaCy is lightning fast, highly accurate and easy to run. It also works well with other tools like TensorFlow, Scikit-Learn, PyTorch and Gensim and provides models for Named Entity Recognition, Dependency Parsing and Part-of-Speech tagging. This library works best while preprocessing data for deep learning. Some of its other features include pre-trained word vectors, support more than 31 languages and easy model packaging and deployment. State-of-the-art speed is the best unique feature and spaCy v2.0 features neural models for tasks such as tagging, parsing and entity recognition.

Code snippet:

```
import spacy
from spacy.tokens import Doc
from nltk.sentiment.vader import
    SentimentIntensityAnalyzer
    sentiment_analyzer =
    SentimentIntensityAnalyzer()
def polarity_scores(doc):
return sentiment_analyzer.polarity_scores(doc.text)
    Doc.set_extension('polarity_scores',
    getter=polarity_scores)
for min tweet: tem =3
doc = nlp(m)
if doc._.polarity_scores["pos"] ==
    doc._.polarity_scores["neg"]: tem = 0
spacy_confused = spacy_neg + 1
elif doc._.polarity_scores["neg"] >
    doc._.polarity_scores["pos"]: tem = 1
spacy_neg = spacy_neg + 1
elif doc._.polarity_scores["pos"] >
    doc._.polarity_scores["neg"]: tem = 2
spacy_pos = spacy_pos + 1
```

VI. Visualization

Data in visual form makes more sense than data in textual form. The representation of data using charts and graphs is more presentable and helps understand facts and figures better.

In this step, we are using statistics and mathematical functions to organize our data in graphical format. Data visualization helps us in understanding changing patterns or trends in the data over time. It also helps in the comparison of the different sets of data.

We are using the following three graphs for basic data visualization:

- Line Plot
- Bar Chart
- Histogram Plot
- Pie Chart

We will do the following analysis and visualize the data in forms of various graphs.

A. Pie charts of our results of APIs

In this, we have visualized how many tweets are classified into which category(positive, negative or neutral) in the form of pie charts for each API.

B. Bar graph of our results of APIs

In this, we have visualized how many tweets are classified into which category(positive, negative or neutral) in the form of bar graph for each API.

C. Comparison of our APIs used

In this, we have compared our APIs and have determined the accuracy of each API that we have used. This accuracy has been shown using a histogram.

- **Sentimental analysis Visualization Results Using various API's**

Pseudocode: To create pie chart

```
# Data to plot
labels = 'Positive', 'Negative', 'Neutral' sizes =
        [text_pos,text_neg,text_confused]
colors = ['gold', 'yellowgreen', 'lightskyblue'] explode
        = (0.1, 0, 0) # explode 1st slice
# Plot
plt.pie(sizes, explode=explode, labels=labels,
        colors=colors, autopct='%1.1f%%',
        shadow=True, startangle=140)
        plt.axis('equal')
plt.title("Sentimental Analysis Using TEXTBLOB ",
        bbox={'facecolor':'0.8', 'pad':5})
        plt.show()
```

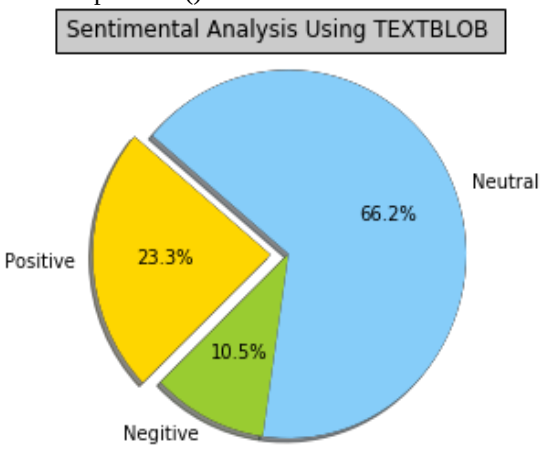

Fig 3. Pie chart of data labelling using TEXTBLOB

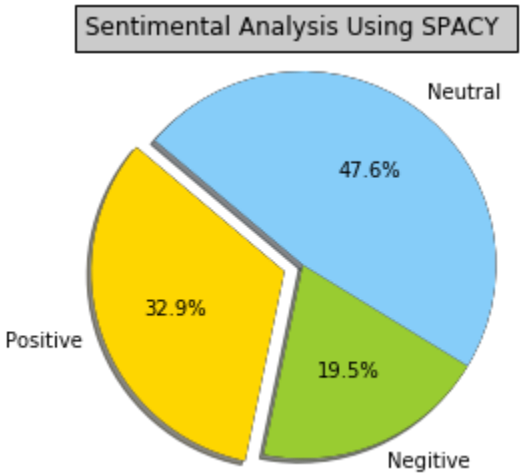

Fig 4. Pie chart of data labelling using SpaCy

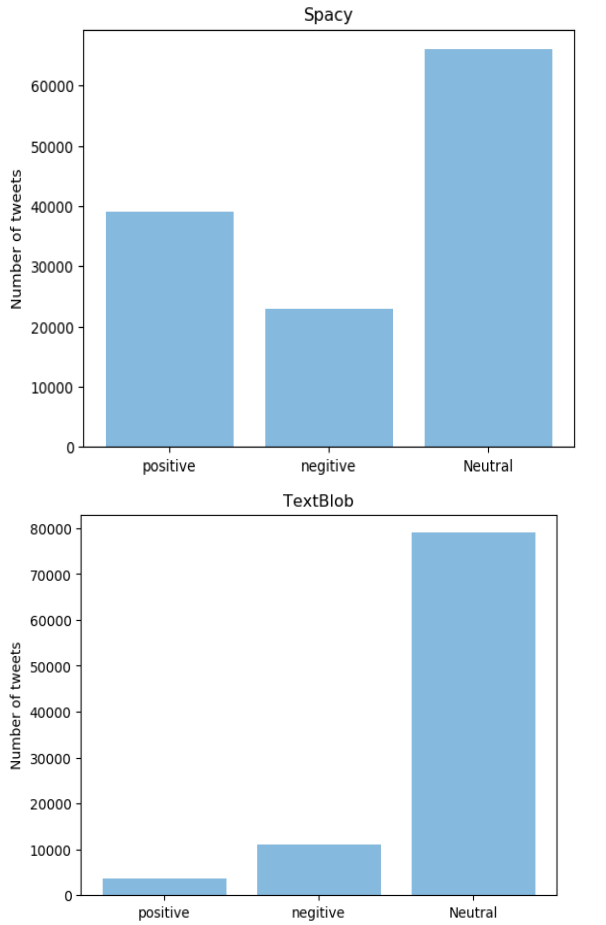

Fig 5. Bar graph of SpaCy and Textblob APIs

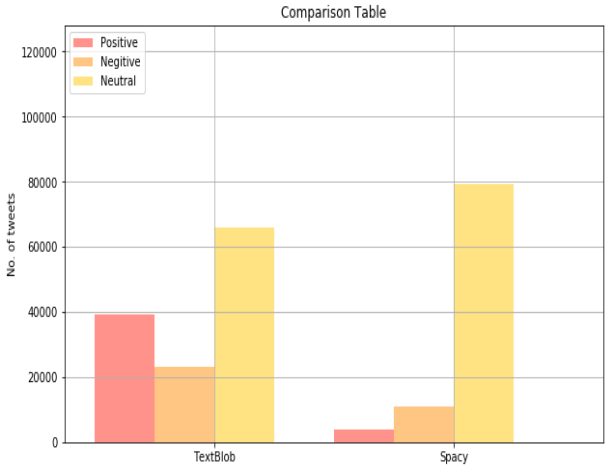

Fig 6. Histogram of SpaCy and Textblob APIs showing comparison

These Pie charts shows Results For SpaCy and Textblob APIs. SpaCy shows better results than Textblob. As neutral is less in percentage. Also, SpaCy took more time than Textblob for analysis. Hence, SpaCy is slow. To get the most accurate and final results we created the text file which contains the result for each individual tweet and we take the mode of both API to check weather the tweet is Positive, Negative or Neutral.

Table 1. Difference between Textblob and SpaCy on the basis of recent Repository

Repository	**TextBlob**	**SpaCy**
Stars	6,130	13,094
Watchers	283	537
Forks	805	2,214
Last Commit	About 2 months ago	2 days ago
Code Quality	L3	-
Language	Python	HTML
License	MIT License	
Tags	Text Processing, Natural Language Processing, Linguistic	Natural Language Processing, Scientific, Engineering

VII. CONCLUSION

In this paper, we used Twitter's tweets on triple talak and analysed how positive or negative or neutral a tweet is. We have used two well-known APIs i.e. Textblob and SpaCy. We compared the results of two API and found that Textblob is faster than SpaCy. But SpaCy produced more accurate results (refer fig 3. and 4.). With gentle learning curve and surprising amount of functionality Textblob has become one of the best language on beginner level.It makes text processing simple by providing an intuitive interface to NLTK. TextBlob can be used for initial prototyping for almost every NLP project.

On the other hand, SpaCy is the new kid on the block, and it is becoming quite sensational. It's marketed as an "industrial-strength" Python NLP library that's geared toward performance. SpaCy is minimal and opinionated, and it provide you with plenty of options like NLTK does. It give the best algorithm for each purpose so we don't have to waste time on choosing an optimal algorithm and we just have to focus on productivity. SpaCy is built on

Cython and lightning-fast but not as fast as TextBlob. SpaCy is known as “state-of-the-art,” but its main weakness is that it currently only supports English.

SpaCy is new but growing at good pace and will probably over take NLTK. If one is building a new application or revamping an old one then one must try SpaCy.

REFERENCES

[1] Taboada, M., Brooke, J., Tofiloski, M., Voll, K., & Stede, M. (2011). Lexicon-based methods for sentiment analysis. Computational linguistics, 37(2), 267-307.

[2] M. Ravichandran, G. Kulanthaivel, and T. Chellatamilan.Research Article on Intelligent Topical sentiment Analysis for the Classification of E-Learners and Their Topics of Interest, Hindawi Publishing Corporation. The Scientific World Journal.Volume 2015, Article ID 617358, 8 pages

[3] Sudhir Kumar Sharma,Sentiment Predictions Using Deep Belief Networks Model for Odd-Even Policy in Delhi.International Journal of Synthetic Emotions , Volume 7 • Issue 2 • July-December 2016

[4] Ana¨is Collomb ,Crina Costea ,Damien Joyeux ,Omar Hasan and Lionel Brunie .A Study and Comparison of Sentiment Analysis Methods for Reputation Evaluation. University of Lyon, INSA-Lyon, F-69621 Vil leurbanne, France

[5] Braja Gopal Patra, Dipankar Das, and Amitava Das.Sentiment Analysis of Code-Mixed Indian Languages: An Overview of SAIL_Code-Mixed Shared Task @ICON-2017.Article · March 2018

[6] Prabu Palanisamy, Vineet Yadav and Harsha Elchuri,Serendio: Simple and Practical lexicon based ap proach to Sentiment Analysis. Serendio Software Pvt Ltd Guindy, Chennai 600032, India

[7] Haseena Rahmath P, Tanvir Ahmad. Sentiment Analysis Techniques - A Comparative Study. IJCEM International Journal of Computational Engineering & Management, Vol. 17 Issue 4, July 2014 25 ISSN (Online): 2230-7893

[8] Mining the Social Web, 3rd Edition.Book by Matthew A. Russell, Mikhail Klassen