Ismail Farooq
01-134162-073
Hamza Alam
01-134162-103

Urdu Sentiment Analysis using Social Media

Bachelors of Science in Computer Science

Supervisor: Ms. Sahar Arshad

Department of Computer Science
Bahria University, Islamabad
February, 2020

Acknowledgments

In the name of Allah, the Most Gracious and the Most Merciful. We would like to Thank our supervisor Ms.Sahar Arshad for giving the opportunity of working our final year project with her. We would like to express our gratitude to our supervisor for being patient and helping us in every possible way. Our supervisor was really forthcoming in completing our project.

Contents

List of Figures

List of Tables

1 Introduction

1.1 Project Background/Overview

Social media is an enormous platform where people from around the world can communicate in an instance. It has affected our personal and business life in such a manner that its waves will ripple through generations to come.
Twitter, founded in 2006 by Jack Dorsey, was initially built as an SMS platform which has now transformed into a micro-blogging platform where vast amounts of data is uploaded everyday by all industries. On average, around 6000 tweets every second, 350,000 tweets every minute and around 500 million tweets are generated day. The popularity of the "hashtag" has to be credited to Twitter even though they are not the ones to create it.

From political topics that decide the faith of a country to miniscule topics are discussed on Twitter. It has become a place of credible source for the journalist community as well. It is no doubt that Twitter plays an important role in our everyday lives. With the vast amounts of data that's available to the public, a question arises, what to do with it? It's humanly impossible to read 500 million tweets a day, and on top of that filter the tweets that are important to the topic one is searching.

The internet is a vast ocean of knowledge, and social media is just an island in that ocean. Although Twitter is just a small piece of the internet we know today, the availability of Urdu text through Twitter will help us get process able corpus. Urdu is the national language of Pakistan and is also vastly spoken in India, Bangladesh and other countries in the subcontinent.

Even though, Urdu is spoken by over a 100,000,000 people, it still lacks any reliable lexical resource. In this project we propose, fetching tweets posted in Urdu of a specific topic from Twitter, analyze the text and give the collective sentiment of users regarding the topic of interest.

1.2 Project Description

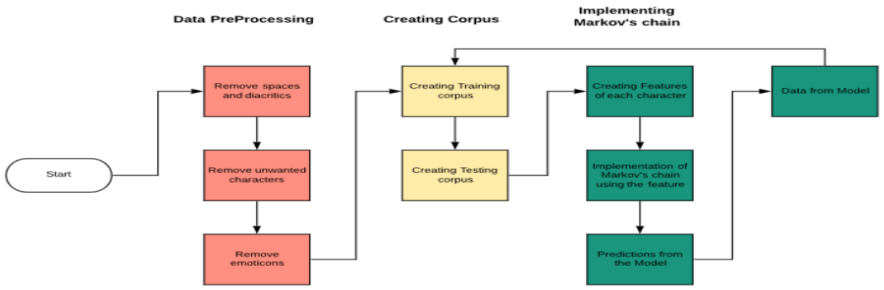

Figure 1: Flow Diagram

In the diagram above, the overview of the steps taken in the project. Initially, steps are performed to process the data before creating test and training corpuses.
After creating corpuses:

- Features are made for each character of the Urdu sentence.
- Using those features of the Urdu text, model is trained.
- Next step involves predicting if the space between two words is a word boundary or a Zero Width Non Joiner.

The output from the model are sentences with correct word boundary analysis and those sentences are used as feedback for testing data to check and see the accuracy of the model prediction.

Research scientists are collecting data from ages and from the last few decades, the data size haven grown massively. Data can be thought of as "raw". It is not in itself any use to us. Twitter is just one of the many platforms that contributes to the ever-growing data.

To make use of that data, we need to process it. The process should convert the "raw" data into useful information. Sentiment analysis is used to detect the emotional tone of a text. It provides understanding of the real mood, attitude and opinion of the writer [1].

Though, there are many reliable lexicons for English, such as WordNet, GeneralInquirer and SentiWordNet, there isn't one for Urdu itself.

For a computer to understand what human beings are feeling, their moods and emotions the computer must apply all its knowledge of the language's syntax, semantics and phonology.

1.3 Project Statement

Unlike English, Urdu is a resource poor language. Most websites in Urdu are created in a graphical format.

So, the lack and unreliability of an Urdu lexicon is a hurdle. Sentence ambiguity is another major problem. For example, "This is not nice". Here the word "nice" is taken as positive but "not" is taken as negative. A solution to the problem is discussed in the Methodology section of the report.

Finally, the use of spaces in Urdu will also become a problem. Word boundaries in Urdu are not defined clearly and so a solution has to be come up with. Many other languages have a similar problem where spaces are used for word boundaries but not consistently.

In Urdu, the use of space in such a manner gives rise to two problems that are space omission and space insertion. Given the tweet, the decision whether the tweet is positive, negative and neutral is an emerging research area. The tweets used will then determine their collective sentiment. The application can be used anywhere for review of brands or products or by political leaders.

1.4 Objective

- To build a mobile application that provides its users visual representation of the sentiment of a specific topic from Twitter for Urdu tweets.
- To analyze people sentiments about a topic of interest.
- To work in our national language as it will be used as an official language in the coming years.

1.5 Scope

We will be working on natural language processing (NLP). Since the context, environment and constraints are different for Urdu language when compared to other languages such as English, the process will involve somehow applications of techniques (text processing, data analysis, data cleansing and prediction) with variations from existing works.

Users, using our mobile application, will be able to get insight on a topic that they can then relate to their business. Our application will give a visual representation of all the positive, negative, and neutral tweets about it.

Processing the tweets in such a manner will allow us to make use of the enormous amounts of data.

People around the subcontinent can learn about the latest trends.

In our competitive markets, sentiment analysis on Urdu tweets can provide many businesses the insight they need in order to increase their Return of Investment (ROI), and consequently be successful in conducting business.

Using tweets that are in Urdu can provide us tailored analysis on current event of the subcontinent.

1.6 Existing System

There are very few websites that perform sentiment analysis on English which uses TWEEZER which is a web application used to analyze the sentiments of the tweets but segmentation is not done as In Urdu sentiment analysis [1].

Click stream analysis also known as click stream analytics which is used in e-business where the data is collected by how many pages are served to the user and how much the user puts the items into the shopping cart or takes out. Large volume of data can be collected by click stream and many e-businesses rely on click stream [2].

The Arabic sentiment lexicon is also a website which uses intensity score for a word or a phrase to predict score using strength of association with 0 or 1. 1 being maximum positive association, 0 being less positive association .

2 Literature Review

Nowadays different languages use different techniques for word segmentation. These techniques are used by researchers from different origins and have deduced significant results. Some of these are discussed as under.

2.1 Word segmentation using machine learning

Word segmentation is a basic Natural processing language task and it plays a role in different NLP areas. IR, POS, NER, sentiment analysis is some of the areas which can benefit a lot from word segmentation [4]. The paper referenced proposes a way for segmentation for Urdu language through machine learning. In this paper, the system intents to adopt the use of conditional random fields (CRF) to acquire the task. Compound words and reduplicated words are also a challenge faced in Urdu text. Machine learning methodology is used in the research paper to overcome the challenges faced in Urdu text.

2.2 Urdu language processing

Western and European languages are full of resources when it comes to segmentation. Variety of tools and linguistic resources are available for them e.g. corpora, WordNet, Dictionaries. On the other hands eastern languages like Urdu are devoid of such. The paper referenced discusses a different linguistic technique available for Urdu language processing. The first step of the paper is that the dataset for Urdu language is discussed. Resources sharing between Urdu and Hindi, Urdu writing morphology and orthography is provided. Preprocessing aspects such as such as stop removal, stemming and normalization are illustrated. Review for the Research for the tasks such as tokenization, sentence boundary detection, parsing, named entity recognition and development of WordNet task are mentioned. Also, the impact of ILP on application areas like information retrieval, classification and plagiarism detection is investigated. In the end, open problems and future directions for the dynamic area of research are provided. The purpose of this paper is to arrange ULP and how it will give a platform for ULP activities in future [5].

2.3 NE Tagging for Urdu based on Bootstrap POS Learning

Part of Speech (POS) tagging and Named Entity (NE) tagging became the main components for the analysis of the text. In this research paper, four levels of text processing are proposed for Urdu model. The system proposed in the paper presents bootstrapping technique applied for the training data of POS learning which improves NE tagging results [6]. The model used overcomes the restrictions imposed by the limited availability of ground truth data that is

required for training a learning model. POS and NE tagging models are based on Conditional Random Field (CRF) approach. In the referenced paper,This model also propose model for boundary segmentation wherever the word written HMM model is trained for character transitions among all positions in each word [6]. The generated words are processed using a probabilistic language model. All of these models use hybrid technique that combines applied models with hand crafted grammar rules.

2.4 Named Entity Recognition System for Urdu

Named Entity Recognition (NER) may be a task helps to find out Persons name, whole names, location names, abbreviation, date and time that classify them into completely different categories that are predefined. This paper focuses on the problems of NER within the context of Urdu language and provides relevant solutions. The system is developed to tag 13 completely different Named entities. NER plays a main role in various natural processing languages fields like Machine Translation, Information Extraction and Question Answering [7].in the referenced paper the system uses the Rule Primarily based approach and developed assorted rules to extract the named entities within the given Urdu text [7].

3 Software Requirement Specifications

3.1 Purpose

To build an application that will allow the users to get an insight on a topic that they can relate to their area of interest. This application will provide a visual representation of all the positive, negative and neutral tweets about it. People can use this application to increase their Return of Investment (ROI) and consequently be successful in conducting business. Using tweets that are in Urdu can provide us tailored analysis on current event of the subcontinent.

3.2 Product Scope

The application that we proposed will provide the collective sentiments of the people using Urdu tweets fetched from twitter using tweepy library. The application will perform word segmentation on Urdu tweets using Markov's chain. After segmentation Urdu to English translation will be done word by word and sentiment analysis will be performed. The application will analyze the sentiments of the Urdu tweets and determine the emotional tone behind the tweets and understand the attitude in which they are written whether they may be positive, negative or neutral. This application will be helpful in business or political views by regard to collective sentiment of the people towards the product or business.

3.3 Overall description

3.3.1 Product perspective

As with the emerging world and ongoing trends around the world our application will help people in decision making weather which product to buy, whom to vote and in which business to invest. This application will be a breakthrough for taking a decision. This application will help business companies understand the social sentiment of the brand and the products and services they are providing while monitoring the tweets.

3.3.2 Product functions

The functionality of the product includes;

- Getting collective sentiments of the people on a related topic.
- Displaying the result.

3.3.3 Operating environment

The application will be downloaded on the user's phone which will be connected to wifis. The user will login into the application, he will type the desired tag in the text box which will display the given results.

3.4 External Interface requirements

3.4.1 User Interface

The user interface of our application will be user friendly. The user can easily navigate through the menu. The user interface will provide with an fast response that will not be time consuming.

3.4.2 Software Interface

Twitter interaction with the user will be involved and it will be done through Markov's chain. React native will be used to develop the front end of the application. Markov's chain will be used for the backend processing.

3.4.3 Communication Interface

The application will be connected to the internet. The user will input the hash tag; the hashtag will be sent to API. The API will process the tweets related to that hash tag and the results will be displayed against that hash tag.

3.5 Requirement Specification

After the problem analysis of system, the current system needs and goals are elaborated and investigated. The functional and non-functional requirements of the application are given below:

3.5.1 Functional Requirements

A functional requirement provides with an overview of the system functionalities or the specific functions that may be needed to accomplish a task. The required functional requirements are as follows:

Preprocessing:

- System should process each new tag and tweets posted.
- Model Formulation.

- System should analyze each Urdu tweet polarity.
- System should be able to predict word boundaries.
- System should translate the Urdu tweet into English.
- System results should be properly validated.
- Results Visualization Synthesis.
- Summary reports should be available in both visual and textual forms.
- Comparison of trends may also be provided.

3.5.2 Non-Functional Requirements

- The installed application should be connected to internet and available 24*7
- The application will be user friendly .
- The application should be able to provide with accurate results of the data gathered.
- The trained dataset should be stable. The Markov chain should work properly without error.

3.5.3 Use Case

Figure 3.1 shows the use case model of the system.

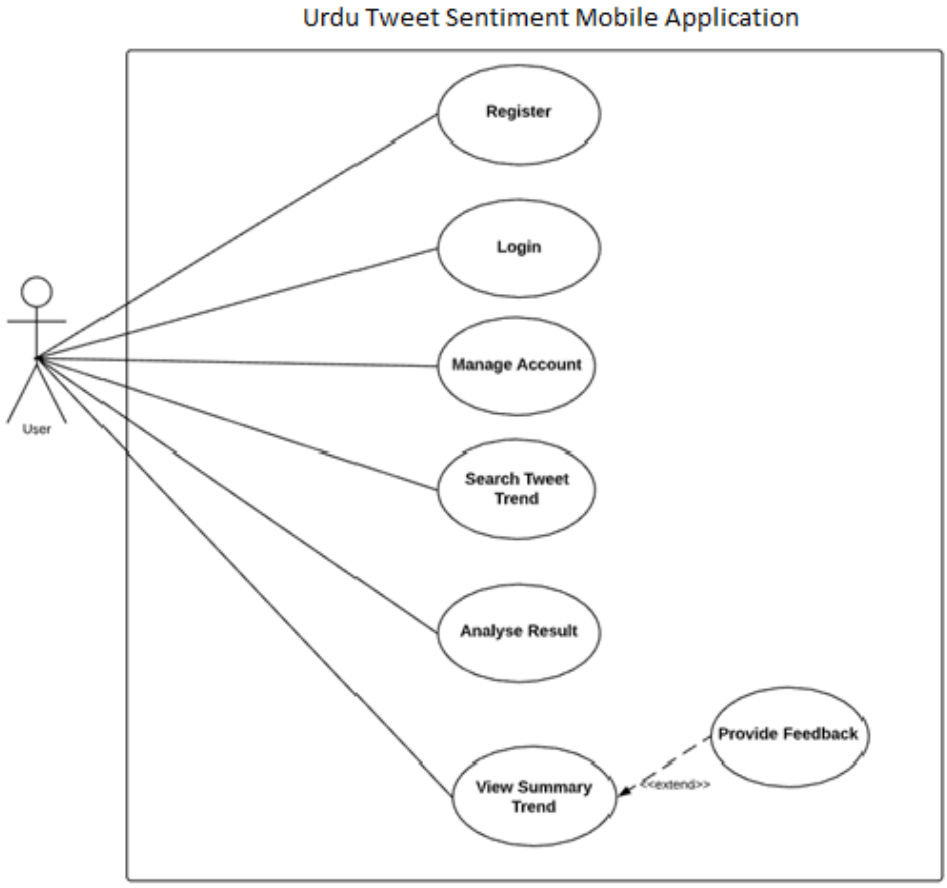

Figure 2: Use Case Model

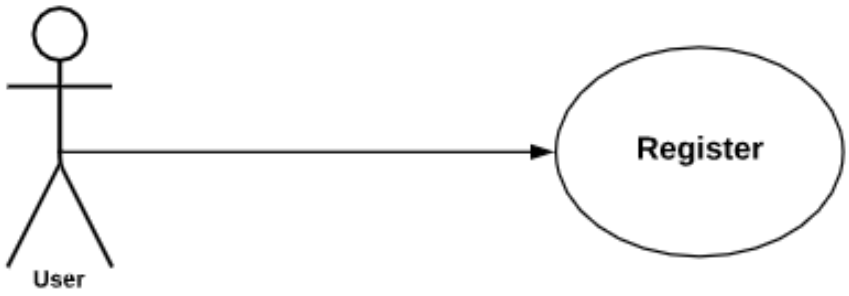

Figure 3: Use Case: Registration

Table 1: Use Case-001: Twitter Registration

Use Case	Register
Actors	User
Description	This allows the users to registers themselves with the application.
Pre-Condition	The user opens the application.
Post-Condition	The user enters the required information.
Basic Flow	The user enters the required information to access the application feature.The application saves the data in the user's account.

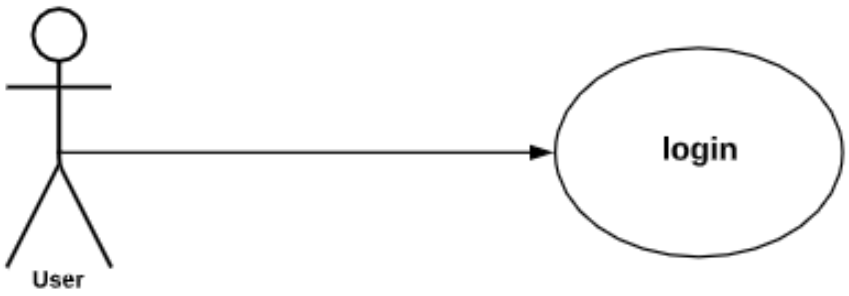

Figure 4: Use Case: Login

Table 2: Use Case-002: Login

Use Case	Login
Actors	User
Description	The user enters the valid username and password to login.
Pre-Condition	The user account is created.
Post-Condition	The user enters into the application.
Basic Flow	The user enters username and password. the system checks the given info if it is valid or not. The use case ends.

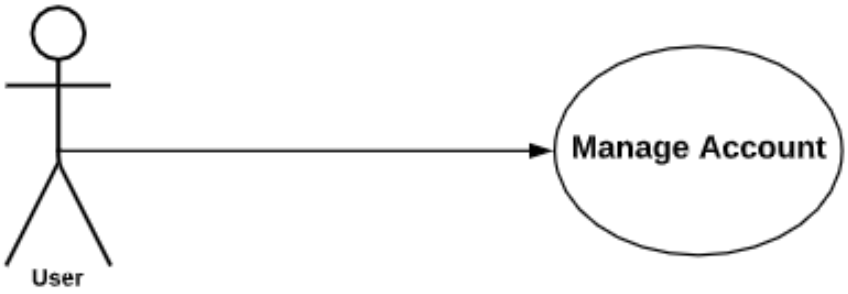

Figure 5: Use Case: Manage Account

Table 3: Use Case-003: Manage Account

Use Case	Manage account
Actors	User
Description	This user wants to add, delete or edit into the account.
Pre-Condition	The user clicks on manage account.
Post-Condition	Changes will be saved after the user hit's the save button.
Basic Flow	The user manages account. Change, edit or delete info. Click on save so the changes may be saved.

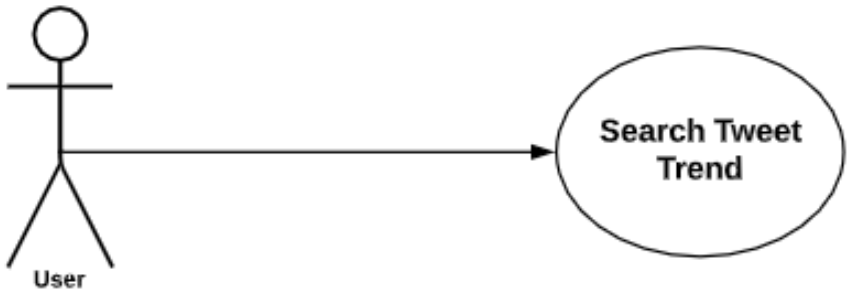

Figure 6: Use Case: Search Tweet Trend

Table 4: Use Case-004: Search tweet trend

Use Case	Search tweet trend
Actors	User
Description	The user will enter the desired tag to search and the result will be displayed on the screen in the form of pie chart or bar graph.
Pre-Condition	The user enters hashtag.
Post-Condition	Sentimental results of the related tweets will be displayed.
Basic Flow	The user opens the application and write the desired tag that will display the result of the related tweets to the tag. The result will be stored for further processing.

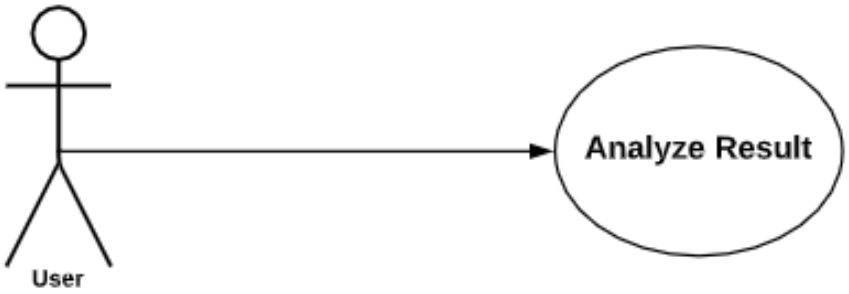

Figure 7: Use Case: Analyze Result

Table 5: Use Case-005: Analyze result

Use Case	Analyze result
Actors	User
Description	The result of the tweets gathered will be displayed to the user after it is calculated.
Pre-Condition	The user enters the tag and click on the analyze result.
Post-Condition	The collective sentiments of the tweets are displayed as either positive, negative or neutral.
Basic Flow	The user searches for the tag and the desired tweets will be displayed which will be further processed for results. The result will then be shown in graphical form.

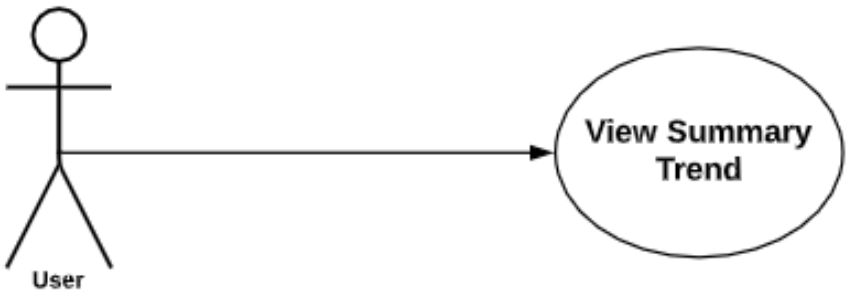

Figure 8: Use Case: View summary Trend

Table 6: Use Case-006: View summary trend

Use Case	View summary trend
Actors	User
Description	This will allow the user to see the overall summary that the user has searched for, the calculated result.
Pre-Condition	The user clicks on the view summary trend.
Post-Condition	The overall trend summary trends is displayed.
Basic Flow	The user clicks on the view summary which displays the results along with what the user history which he has searched.

4 System Design

This chapter contains the development and design phase of the project where the system architecture is discussed in detail. It contains figures, modules, high level and low-level designs of the application that would be helpful in determining the project.

4.1 System Architecture

In system architecture the different design components of the sentiments analysis application will be discussed in detail. This will help in understanding the overall behavior of the system and structure. This will give an over view of the system which will be useful in making of the project.

Our system is a two-tier architecture mainly presentation layer and logical layer. Presentation layer contains the contents of the user interface which is responsible for delivering important system information to user interface. While logical layer is backend development of the application in which developer interaction is involved.

4.1.1 Presentation layer

The language used to design the front end of the application is React Native. The user interface will consist of a textbox through which the user will search for the Hash-tag and the application will fetch the related tweets and process it. The end result will be shown in either a pie chart or a bar graph.

4.1.2 Logical Layer

The logical layer consists of the backend processing of the application. The tweets will be fetched using tweepy library, once the text Is fetched, word segmentation will be performed which will be done by Markov's chain. After the segmentation is complete each Urdu word is translated into English. The text will then be analyzed by SentiWordNet (NLP) which will return the result to the application layer. Hence this layer is responsible for the actual working, performance and behavior of the system.

4.2 Design Methodology

In design methodology necessary tools and techniques are required to design the sentiment analysis application. There are certain procedures which are as follows:

- Fetching the tweets will be done using the Tweepy library.
- Once the text is fetched, we will need to perform word segmentation on the text.
- In word segmentation, we divide and the text into units that are own their own meaningful.
- We will be performing word segmentation using the Markov Chains.
- Markov's chain will be used to model the system that will be trained on an Urdu dataset.
- Based on this model, our system will be able to predict word boundaries (word segmentations).
- Validation for word segmentations will be done using confusion matrix, also known as the error matrix and by using accuracy, recall, F1 Score etc.
- After word segmentation is complete, each Urdu word is translated to English. Translating the text will allow us to use well built, reliable lexicon used for English text sentiment analysis. This process will be accomplished using an Urdu to English library.
- The tweets are filtered for specific words. Some ambiguous words will be filtered-out to increase the accuracy of our model.
- The filter result will then be analyzed by the SentiWordNet (NLP). SentiWordNet is an English sentiment analyzer that uses WordNet, a lexical database.
- The validation of the polarity of the tweets will be done using user feedback.
- Finally, using the data returned by SentiWordNet, we will present the data to the user. Front-end development will be done using the React Native framework.

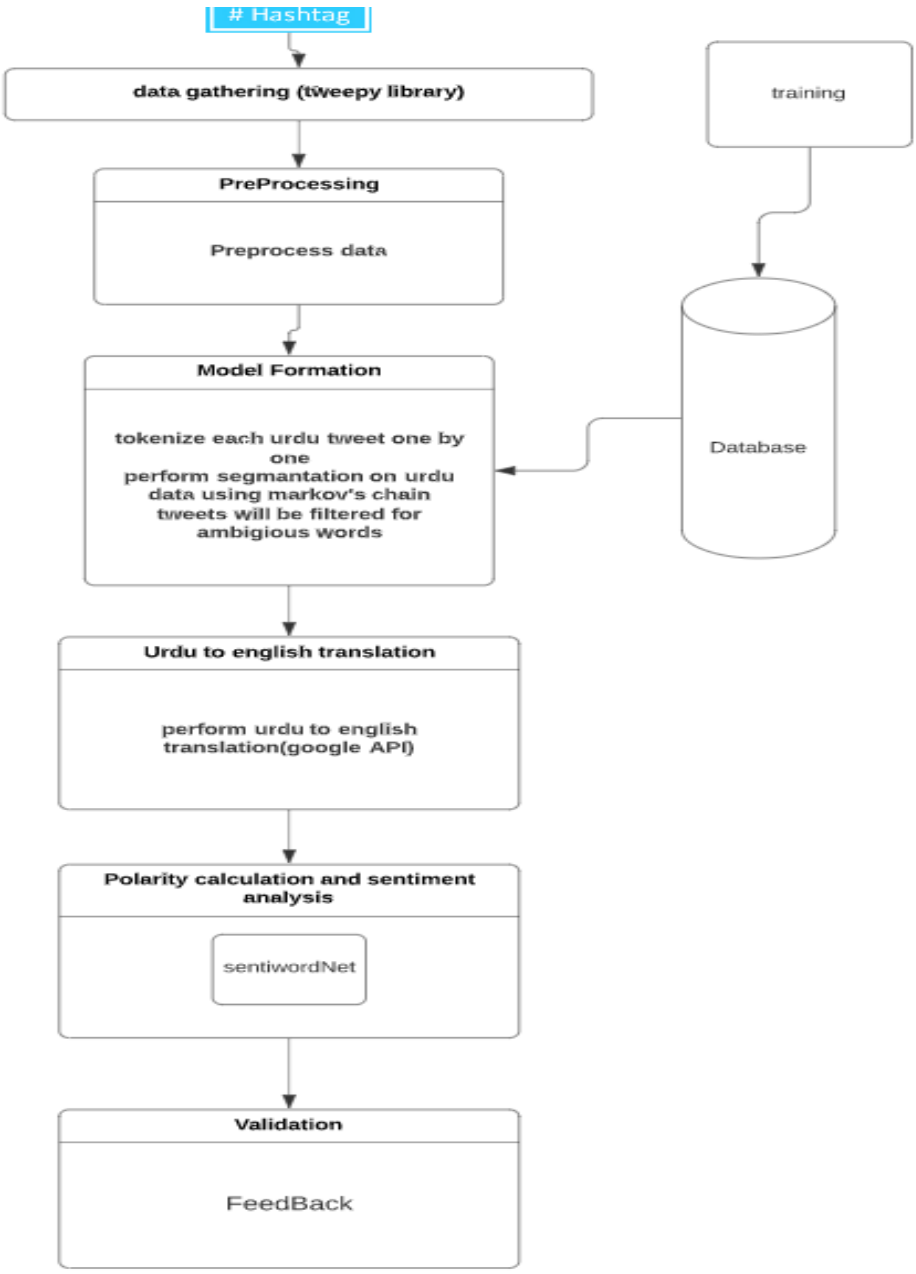

Figure 9: Design Methodology

This figure of Design methodology shows how the data gathered is being pre-processed. To model the system Markov's chain is being used. The resultant

data is then translated from Urdu to English, afterwards to check the polarity sentiment analysis of the resulted data, sentiwordNet is being used which gives us the final feedback.

4.3 High Level Design

The high-level design of the application will cover the relationship between modules and sub modules. How these modules are connected and how they interact with each other. It will include the description of how the overall system will work thoroughly.

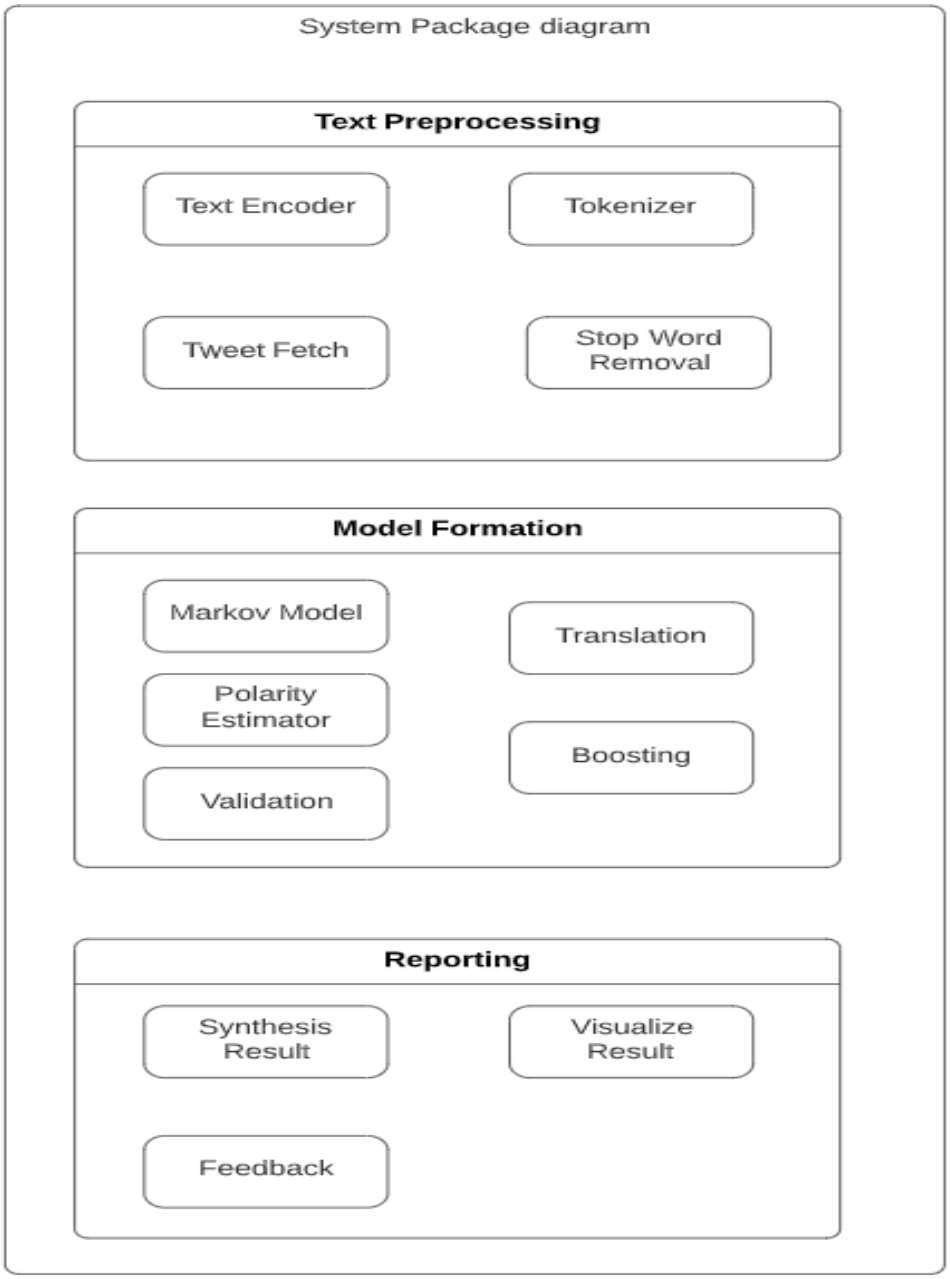

Figure 10: High Level Design

As you can see from the figure the project is divided into three parts; Text processing, Model Formation and Reporting. In text processing; text encoding

takes place to encode the fetched tweets. Then the fetched tweets are tokenized and the stop word removal takes place which removes the stop signs.

In Model Formation; Markov Model is used to process the tweets. Once it is done the polarity of the processed tweets is checked. afterwards translation of the Urdu text to English takes place and then text is validated.
In reporting section; the result is calculated and displayed in the visually which gives us feedback.

4.4 Sequence Diagram

4.4.1 Search tweet trend

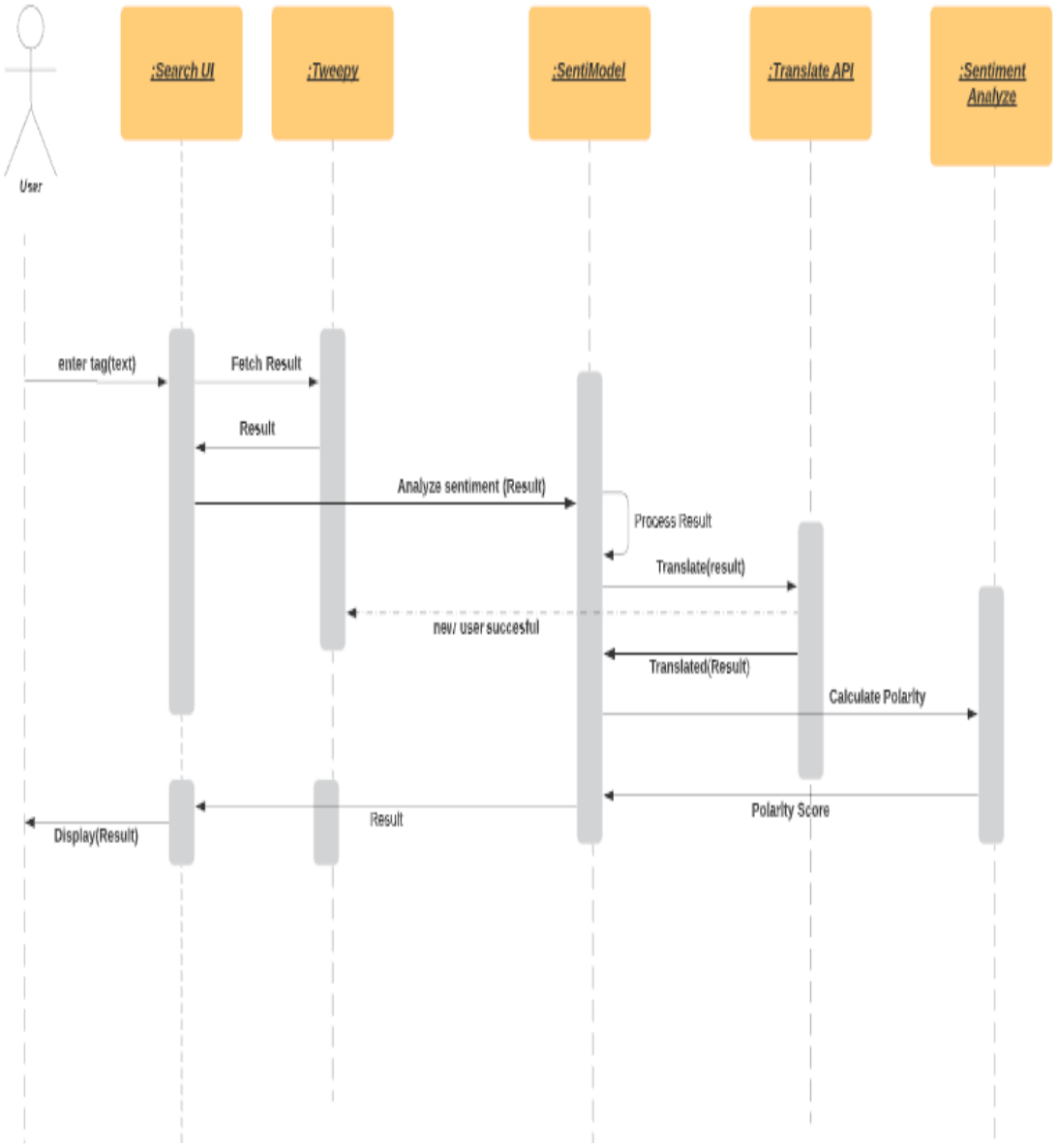

Figure 11: Search Tweet Trend

4.4.2 Login

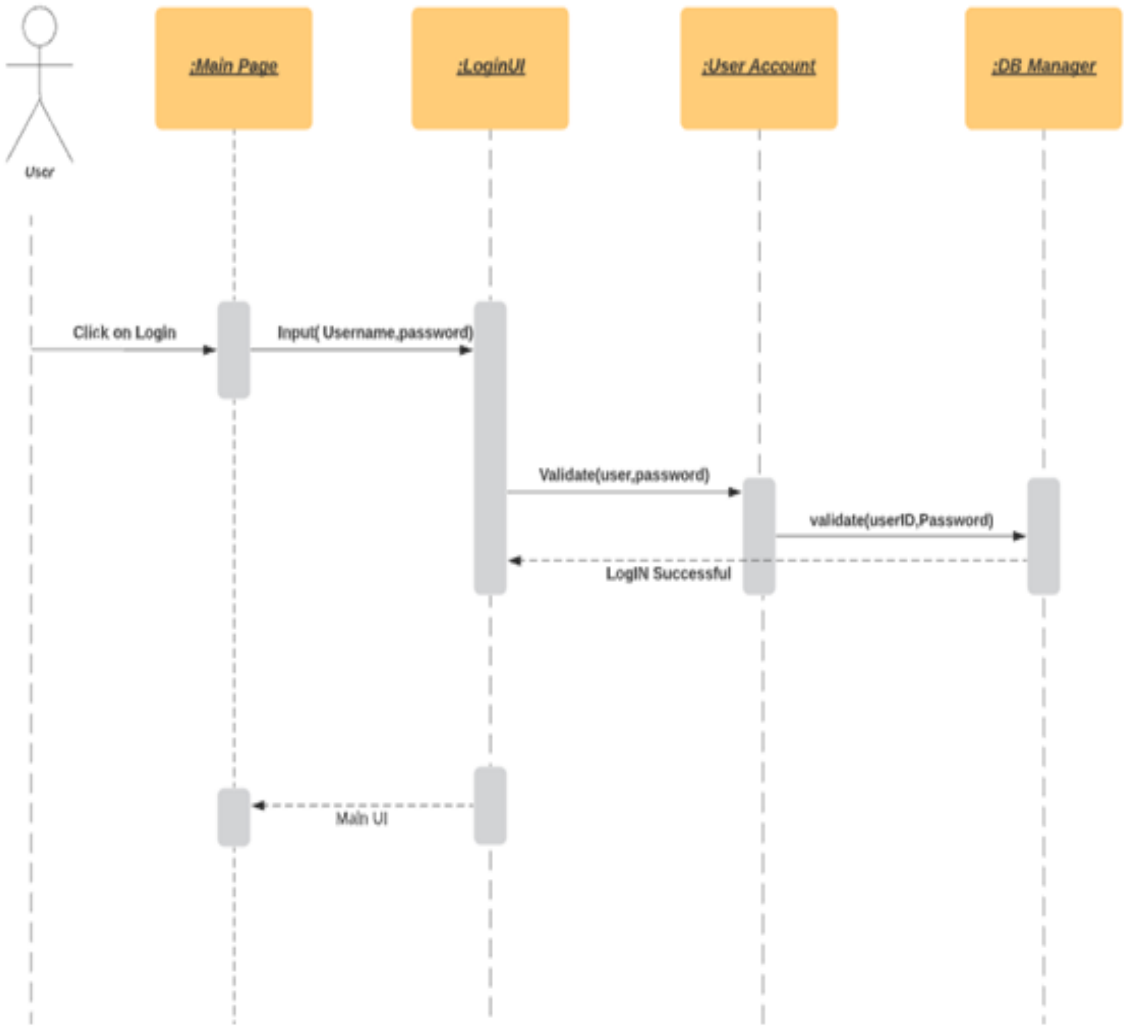

Figure 12: Login Sequence Diagram

4.4.3 Registration

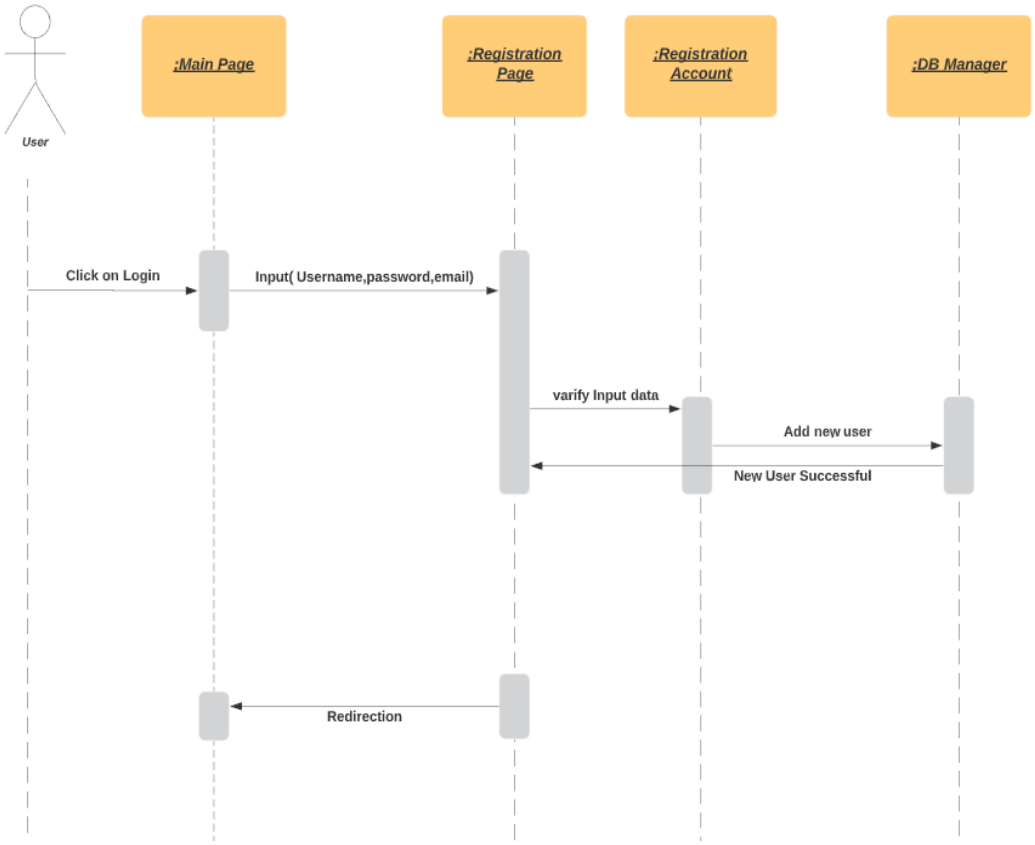

Figure 13: Registration Sequence Diagram

4.5 Activity diagram

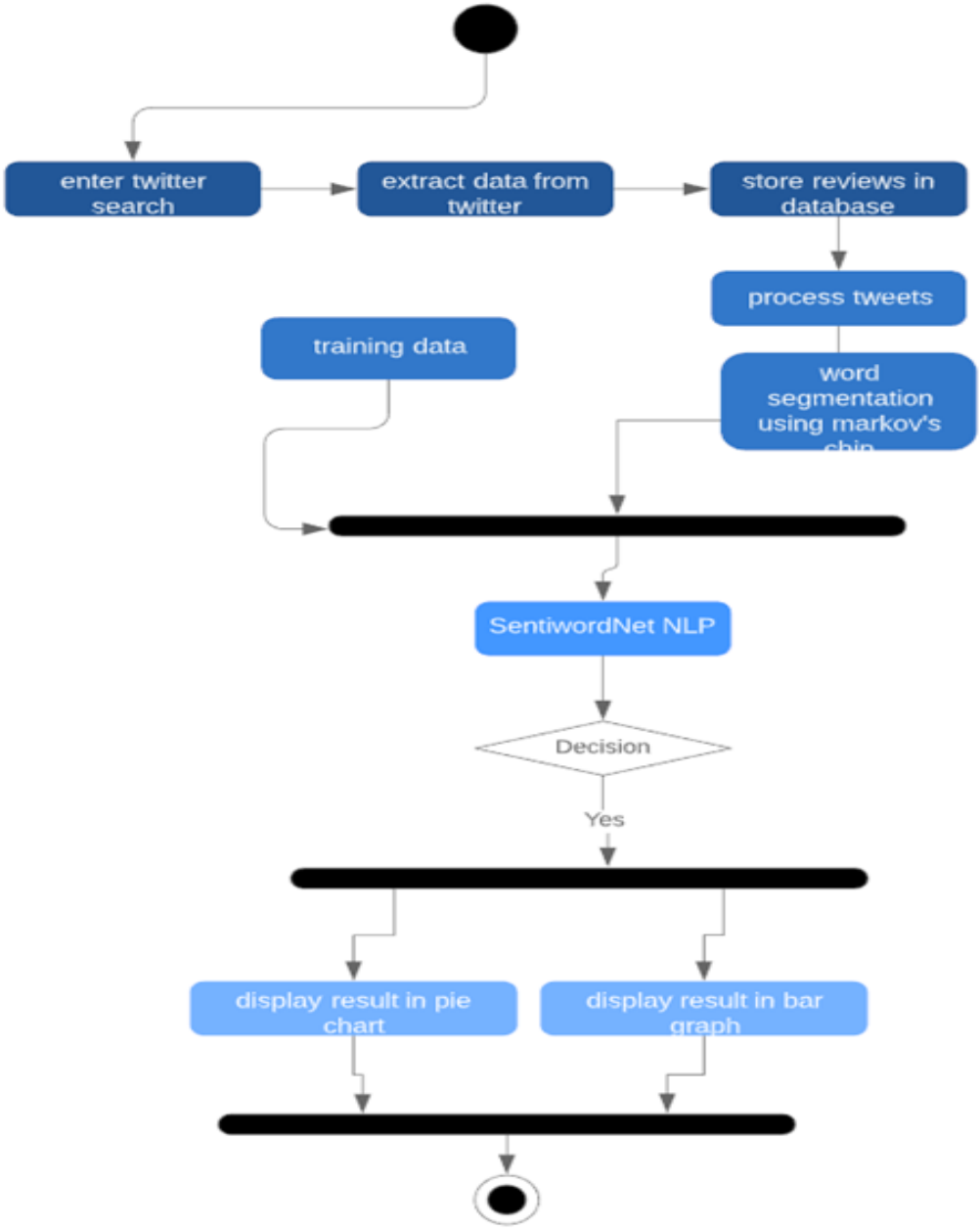

Figure 14: Activity Diagram

The given figure of activity diagram shows how the tweets that are being searched are extracted and stored in a database for further processing. The processed tweets then goes through Markov's chain for segmentation and a result is generated by the trained model. The result is then displayed in Pie chart.

4.6 GUI Design

The graphical user interface of our application is very simple and easy to use. The user will have no difficulty in operating it. The user will on search the tag in the text box and results will be displayed him in bar chart or pie chart.

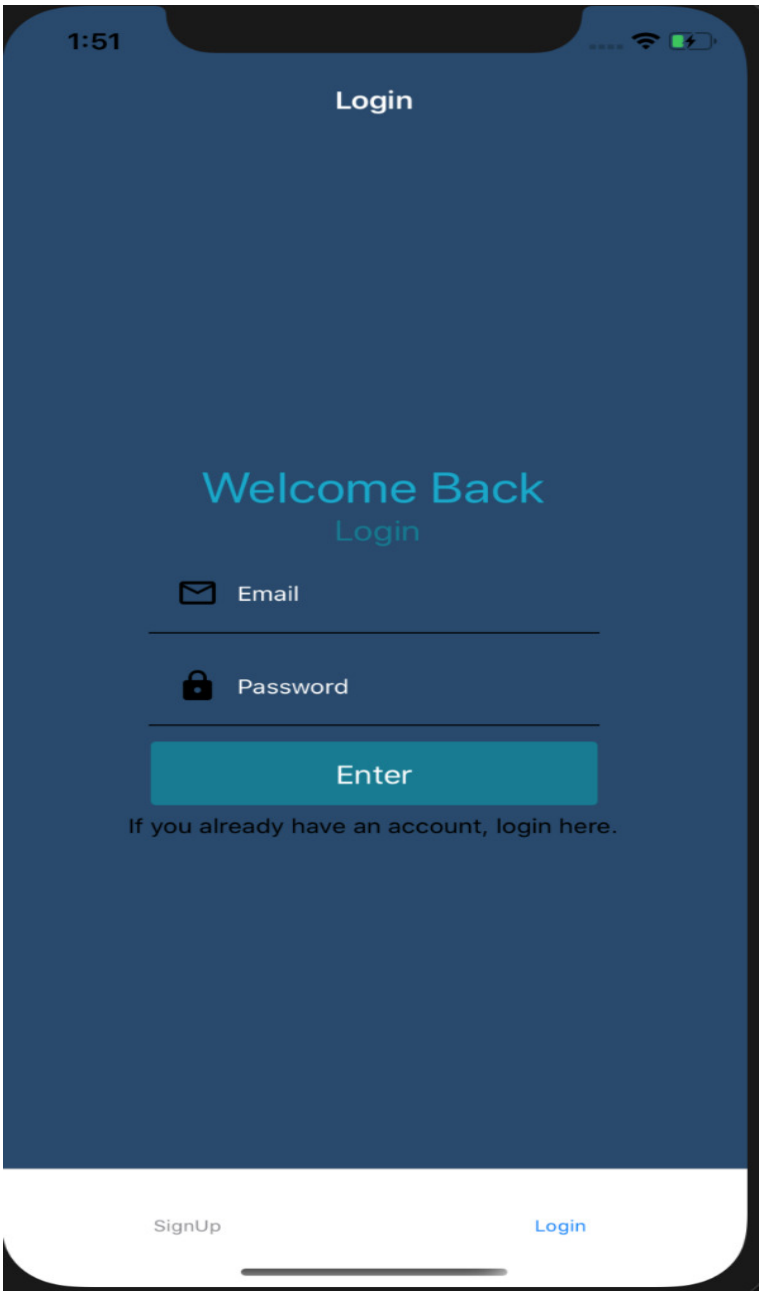

Figure 15: User Interface Diagram of Login Scenario

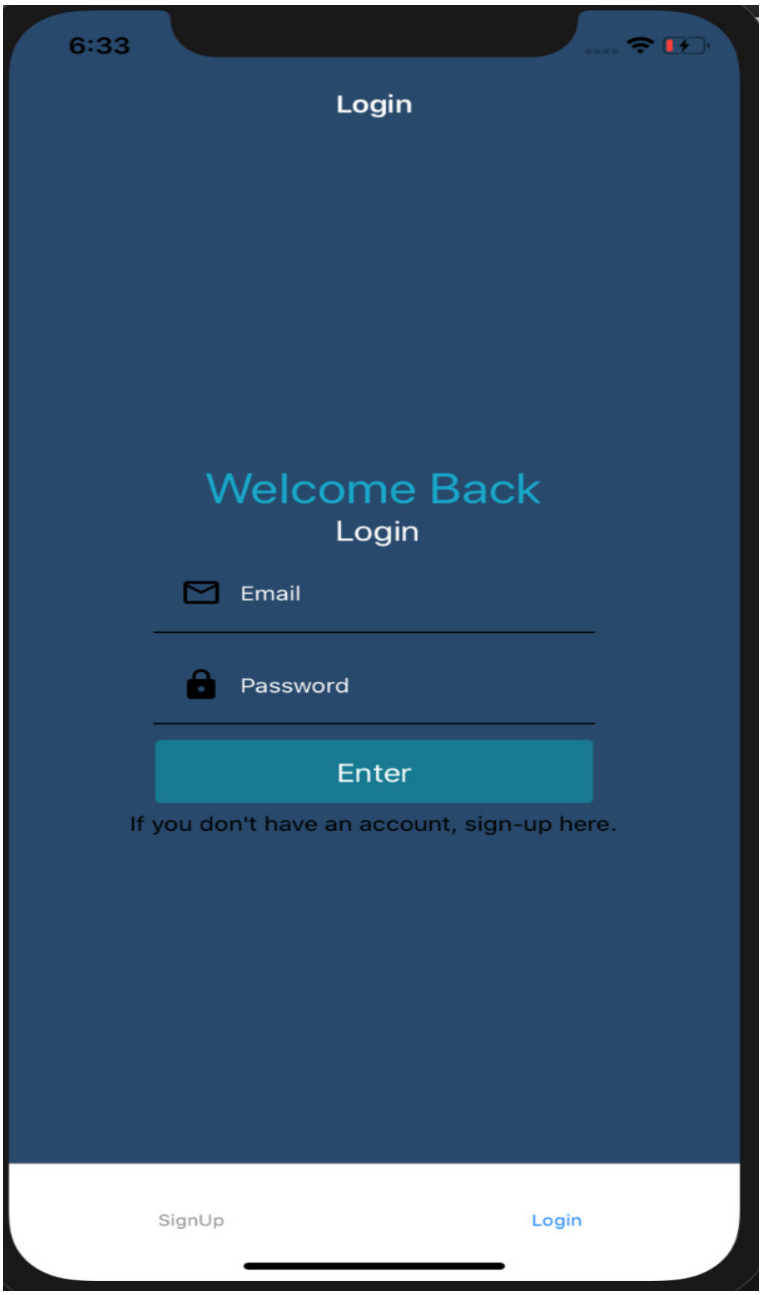

Figure 16: Sign Up Activity

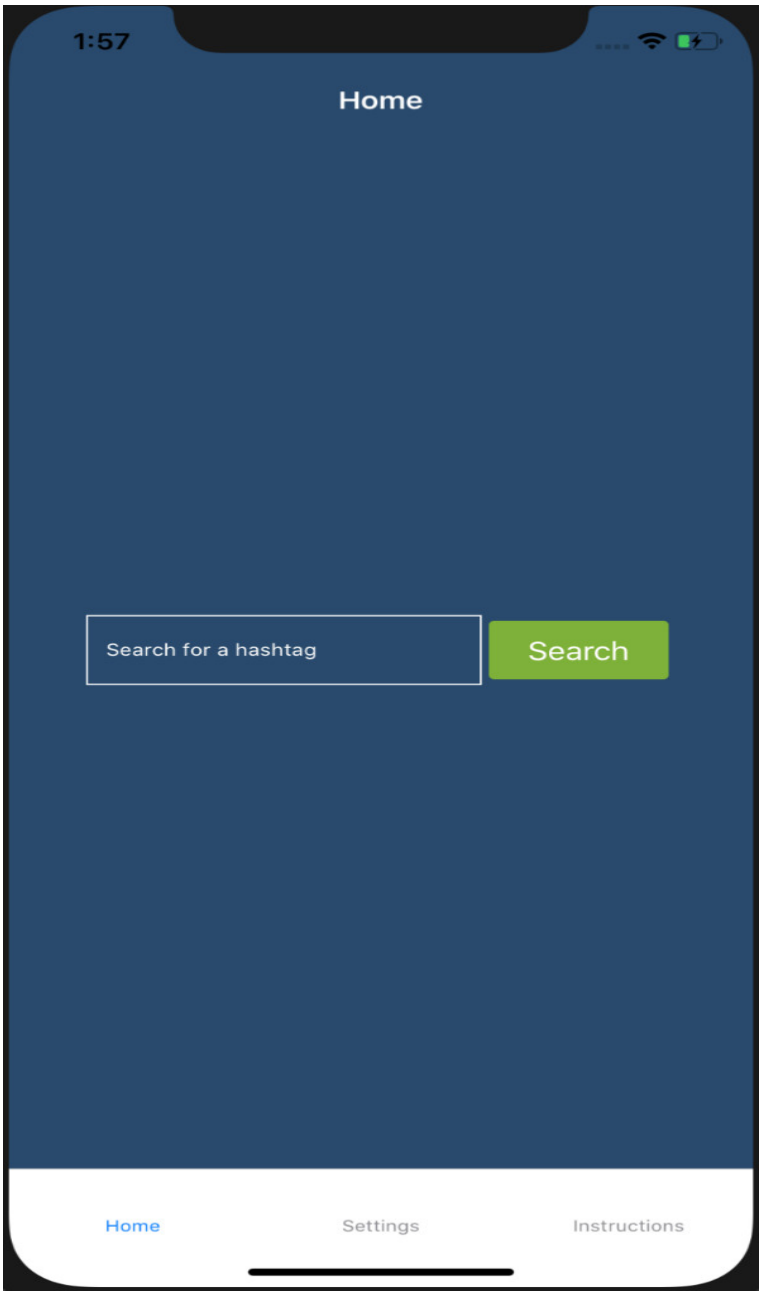

Figure 17: Search Hashtag

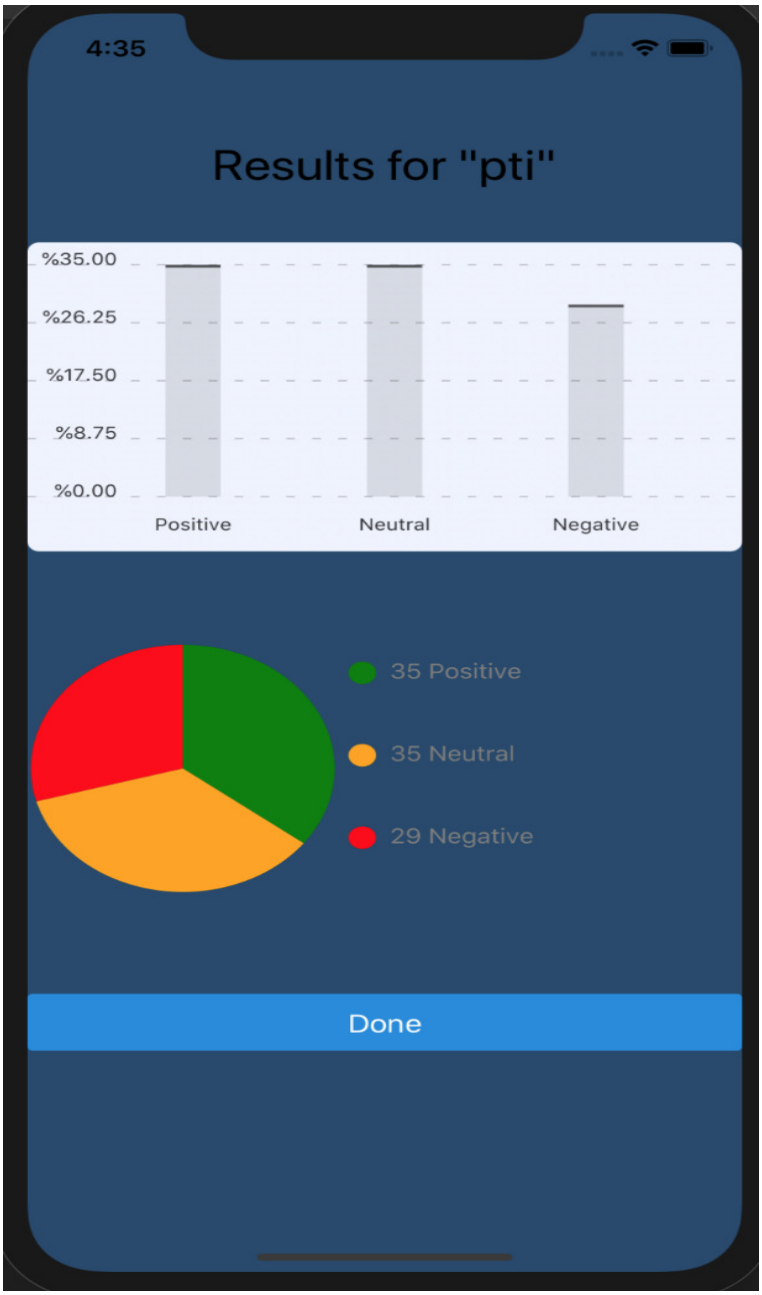

Figure 18: Result

In figure 18, the search results of current ruling party are presented. Since people opinion may change over the passage of time, there are around equal

number of positive and negative reviews about the current governance. These search term, as an input from mobile device, is passed to our server where all relevant twitter Urdu data is parsed, translated into English and then overall parity estimation was performed.
We ran a querry on PTI which gave us the positive, negative and neutral sentiments of the gathered data.

5 Implementation

5.1 Graphical user Interface

The graphical user interface of the application contains sign-in and sign-up options. Once the user is logged in, a Text Box will appear in which the user will enter the tag to search and a click button to initiate the process.
A loading page will be displayed while the user waits for the API response.
The results are shown in a Bar chart and a Pie chart that shows the percentage of people with positive, negative and neutral response to a tag.

5.2 Markov's chain

A Markov's chain is a process in which probability of the next state's dependency is on the attained previous state. The predictions of the future states can be made using this process. It is known as Markov's process. In simple words the predictions on the next and previous state can only be made on present state.

When a text based Markov's chain is generated:

- The body of the text is split into tokens (words, punctuation).
- A frequency table is built. It's a data structure for every word for every word in your body of text; you have an entry (key). This key is mapped to another data structure that is basically a list of all the words that follow this word (the key) along with its frequency [8].
- The Markov's chain is generated. A starting point is selected (key from frequency table) and a random state is selected. The next word that is chosen, its dependency is on the frequency of the previous state. That word is then chosen as the new key and the process is repeated

Usually the transition of one state to another is probability based. In the above mentioned text-based Markov's chain, the probability of transition is based on the frequency of the words and the selected words. The frequency table represents the possible successive states and the selected words represent the previous states. If you know the previous state you will find the successive state. This is the only way to get the right frequency table.

In this project Markov's chain is used on Urdu text. Urdu segmentation can be done using Markov's chain. English segmentation is easy but Urdu segmentation can be a challenging task. Urdu is a resource poor language. There are already developed tools for other languages like English but not for Urdu which makes it challenging. This will allow the Users, using our mobile application, will be able to get insight on a topic that they can then relate to their business. Our application gives a visual representation of all the positive, negative, and neutral tweets about it.

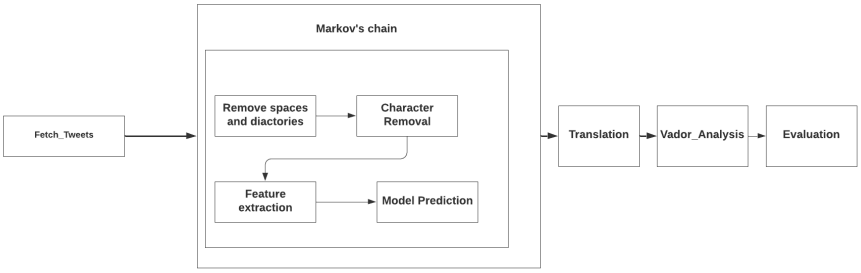

Figure 19: Block Diagram

As you can see from the diagram above, after getting the data from the twitter, we clean the data. Since the tweets contain some English words such as any mentions or tags, we pre-process the data. We also remove diacritics and spaces from the Urdu text as well as any emoticons used in the tweets so at the end we have only the Urdu text.

Next, for each alphabet in a sentence, we create a set of features and states. Features of each character includes:

- isDigit
- isnonjoiner
- category
- direction

The states of each character is also saved as features creating a trainable/predictable Markov chain.
Now, with the list of features we feed it to our model. The model predicts which of the spaces between the words are word boundaries and which are zero width non-joiners. These predictions are made through a model that is trained using a data set that contained over four thousand (4000) reliable sentences of Urdu language. Out of those 4000 sentences, 825 sentences were separated for the sake of testing and the rest where used for training the model.

Using this way, each word in every sentence is labeled to either word boundary or a zero width non-joiner and this way, word boundary analysis is done.

Once labelling is done, next, we translate the Urdu text into English text. Translation of Urdu to English text is done using Google Translate service.

Next, our translated text is passed through Vador Sentiment Analyser. Each sentence is assigned 4 values which are positive, negative, neutral or compound. From these 4 values, we use the compound value to calculate the sentiment of the sentences; that is if the sentence is positive, negative or neutral.

Given below is the Algorithm

```
Begin
        DP (document polarity)=0
        SP (sentence polarity)=0
        Num-S=0
        For each Sentence S belongs to LS
                For each U-W belongs to S
                Translate U-W to E-W with the help of UE-Dict
                If (EW exists in SWN)
                Then {
                Compute sentiment polarity of E-W
                }
                End for each
                Compute sentence polarity
                Add SP to DP
                Num-S=Num-S+1
                End for each
                Compare DP
                If DP> Threshold
                        DP=+ve
                Else
                        DP=-ve
                End it
                Return DP
End
```

Figure 20: Proposed Algorithm

Given below is the figure of Markov's chain feature

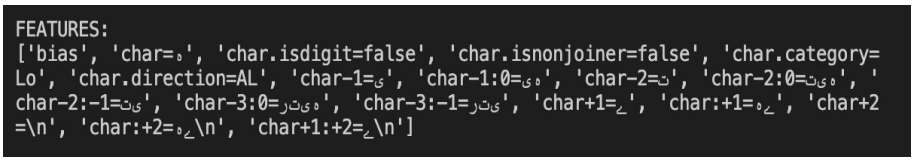

Figure 21: Markov's chain features

This figure states the properties of the each character. For e.g '' predictions is made on the next 3 digits weather the word will be with this character or not. This algorithm predicts the previous 3 digit attached with the character, that's how all possible combinations of previous and next states are made. 'char-1' predicts the previous digit of the current state,'char-2' predicts the last 2 digits and 'char-3' predicts the last 3 digits in the given figure.

5.2.1 MODEL

Markov's chain is used to model the system that will be trained on an Urdu dataset. Once it's complete, the user operates the application where he would write a hashtag and the desired tweets will be fetched using API. Those tweets are then preprocessed. Word segmentation is performed on the Urdu data using Markov's chain. The tweets are filtered from ambiguous words, hashtags, operators, symbols and brackets, this filtered data is then trained using feedback.

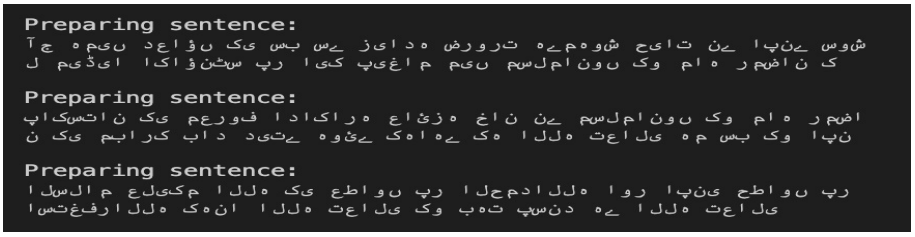

Figure 22: Sentence Preparation

Preparation of the sentences takes place here; all the spaces are replaced and Labels are generated, means when sentence is transferred into characters so every character becomes an index of an array. Now we have list of character in which each alphabet is separated.
Now each character will be labeled 0 and 1 will be labeled to space.

5.2.2 Evaluation

Feed back is done twice in this project.

Once, to improve the model itself for segmentation by feeding our model with data we fetched from twitter. Once the Urdu text is fetched and cleaned for any unnecessary characters, we create character features for each alphabet of the sentence. Eventually, we obtain text that can be used to train the model and improve it.

The other process where feedback is involved is during sentiment analysis to check for polarity.
Feedback from users is taken after results of query have been displayed. Users will rate the sentiment polarity with their best of knowledge and also type in why they have given such a rating. This will allow us to differentiate between good and bad polarity predictions.

Confusion matrix is also known as error matrix, it's used to validate the accuracy of the word segmentation. F1 score is generated by confusion matrix which will determine if it is worthy to put in the feedback. Then again the model is trained which gives next predictions. Over the time the accuracy of the model will increase and it will give better results.

	precision	recall	f1-score	support
I	0.94	0.95	0.95	303
Bw	0.84	0.78	0.81	97
Bs	0.00	0.00	0.00	0
accuracy			0.91	400
macro avg	0.59	0.58	0.59	400
weighted avg	0.92	0.91	0.91	400

Figure 23: Confusion Matrix table

To calculate the performance of the algorithm, confusion matrix is used. The row in the table of the matrix represents actual class instances and each predicted class is represented by column. Confusion matrix removes the confusion as the name states by making it clear for us what kind of confusion takes place in the classification algorithm.

The accuracy in the given table is not sufficient to calculate the performance. Suppose we have 100 samples where 95 of the cases are negative and 5 of the cases are positive. Suppose it is run through a classifier, it will be recorded as negative. The accuracy given out will be 95 percentage because the positive cases were not recognized.

The algorithm labels are calculated by precision and recall. Recall is also known as true positive rate. It is used to calculate the positive results of the labeled words where accuracy is used to calculate True negatives. The precision and recall are calculated by the formulas given

$$\text{Precision} = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

$$\text{Recall} = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

F1 score is a way of calculate precision and recall. F1 score is a best way to balance recall and accuracy. F1 score will give it's correct accurate value if both precision and recall are at the value of 1. It is calculate by the formula

$$F1 = 2 \times \frac{Precision * Recall}{Precision + Recall}$$

```
Training # of Sentences:  4325
\Testing # of Sentences:  100
```

Figure 24: Testing 100 lines of Urdu

Accuracy testing of the segmentation is done through confusion matrix. The text for testing that is used is of 100 lines of Urdu text and that is tested against the model. The result of this process is Confusion matrix.

5.3 Google Translator

Each Urdu word is translated to English after segmentation. It is done using Google API. Its translation will allow us to use English text sentiment analysis. The process will be carried out using Urdu to English library.

5.4 SentiWordNet NLP

SentiWordNet is used for Polarity calculations and sentiment analysis. The translated English text will be analyzed by SentiWordNet. Polarity will be checked by user's feedback. The resultant data will be presented to user in the form of graphs.

5.5 Tools and technology

React native: React Native framework is used in the project for the building of the user interface. It uses JavaScript library for the development of GUI. The user will login/Signup, he will be displayed a textbox in which he will enter the desired and the desired tweets will be fetched against that tag. All the twitter tweet analysis data fetched will be showed through react native which will show data in the positive, negative or neutral in the form of pie chart or bar graph.

Flask: Flask is a lightweight web framework written in python. It is easy to start the application with flask because it does not require any particular tool and its dependency on the libraries is very little. All the processing of text analysis, Markov's chain, and translation is done in this framework which generates our API.

Postman: postman is used for API testing and is one of the popular tools used widely.
Emulator is used in testing applications and games on as many as devices before it's product is launched.

5.6 Environmental languages used

JavaScript : JavaScript in the project is used to call the API and display the result. Modern JavaScript makes it easy to build complex application using React native. JavaScript is used to build the react native application of the project and for API calling.

XML : Extensible markup language (XML) is used for the development of graphical user interface which is developed by Microsoft. This is used in react Native for the front end design to create Graphical User Interface. It is the essential part as the user will interact with the application through GUI. It's used for the components used in the UI. React native components are XML components which is used in the application.

Python : Python is a programming language used in the development of backend server. The server side API is in Python language. Python in the project is used to fetch the data from twitter using API. The development of the project algorithm is done in python such as gathering of tweets, preprocessing of text, Markov's chain, calling of API's, Urdu to English translation, validation of the texts.

6 System testing and evaluation

6.1 Graphical user interface

Graphical user interface testing is done to ensure the desired controls on the screen are working properly such as buttons, textboxes, icons and dialogue boxes. Graphical user interface testing is important because it involves direct interaction with the user. The objective of the application is that the user with no knowledge should be able to understand and easy to use the application. Making the text boxes easy in such a way that user will only enter a desired tag and the graphical results will be displayed to him, making it easy for him to understand.

6.2 Usability testing

This testing is done to ensure that the new user with little knowledge will be able to operate the application without any difficulty. The user will not face any problems while logging in to the system. The user will simply write the desired tag and he will be able to get the graphical result.

6.3 Compatibility testing

Compatibility testing ensures the platform on which the application can be installed. Our application will use both android and iOS based; application is compatible on both iPhone and android.

6.4 Application performance testing

Application performance testing is carried out to for non-functional requirements of our application. It is used to measure the quality attributes that makes up the system.

6.5 Installation testing

Installation testing deals with the installation process and installation device. Our application can be installed on both IOS devices and android devices. The user must download the application from App Store and Play Store on his device.

6.6 Load testing

Load testing deals with the number of tweets our system can fetch at a time. Our application fetches around 300 tweets within couple of seconds. These 300 tweets are then be processed using Markov's chain and a collective sentiment

result will be displayed.

6.7 Test cases

Table 7: Test Case-001: Application installation

Test Case ID	TC-01
Test Case Title	Installation of application.
Description	The process in which application is installed.
Test Steps	1. The user should be connected to the internet. 2. The user should open App Store/Play Store. 3. The user should download the application.
Expected Result	the application is installed on device.
Status	Pass

Table 8: Test Case-002: Application running

Test Case ID	TC-02
Test Case Title	Running the application.
Description	The application is run by the user.
Test Steps	1. 1) The user should tap on the application icon. 2) The user should enter the desired tweet tag. 3) The desired results should be displayed to the user .
Expected Result	The application is operatable on the device.
Status	Pass

Table 9: Test Case-003: Graphical user testing

Test Case ID	TC-03
Test Case Title	Graphical user interface testing.
Description	The GUI of the application is tested user.
Test Steps	1) The user should open the application . 2) Navigate through the menu back and forth . 3) Check if the application if its working correctly.
Expected Result	The application is easy to understand and the GUI is working.
Status	Pass

Table 10: Test Case-004: Application performance testing

Test Case ID	TC-04
Test Case Title	Application performance testing.
Description	Effectiveness of the application is tested.
Test Steps	1) The user should run the application . 2) Run the desired tags 10 times and repeat. 3) Evaluate the result.
Expected Result	The application results are accurate.
Status	Pass

Table 11: Test Case-005: Compatibility testing

Test Case ID	TC-05
Test Case Title	Application Compatibility testing.
Description	To verify the compatibility of the application with devices.
Test Steps	1) The user should open the application . 2) See if the application is running smoothly with the device. 3) The user should open the application on another device.
Expected Result	The application working smoothly on devices.
Status	Pass

Table 12: Test Case-006: Sign Up Testing

Test Case ID	TC-06
Test Case Title	Sign Up.
Description	The user will create an account on the application.
Test Steps	1) The user will enter his credentials . 2) The user will press the Sign Up button. 3) The user will be Registered.
Expected Result	The user will be signed up.
Status	Pass

Table 13: Test Case-007: Search Tweet Testing

Test Case ID	TC-07
Test Case Title	Tweet Search.
Description	To search for the desired tweet for the sentiment analysis.
Test Steps	1) The user should open the application . 2) The user will enter the related tweet tag. 3) The user will be displayed with graphical result.
Expected Result	The application will display the result graphically.
Status	Pass

7 Conclusion

Developing Urdu tweet sentiment analysis is a new and challenging task. Little work has been done on Urdu segmentation. Our application will be able to fetch tweets regarding a topic and display its result in positive, negative or in neutral manner. This will help people doing business or it would help people in voting. Urdu being the national language which is widely used along with twitter, an enormous micro blogging network together can benefit a lot of people. Our application will be able to provide analysis of the related topic which will help the user in decision making. Urdu being a resource poor language, the availability of the Urdu lexicon was a challenging task. Word boundaries in Urdu are not defined. The use of space was all a problem in Urdu which made our project a bit slow.

8 Bibliography

References

[1] Kumar, N., 2017. Twitter Sentiment Analysis. [online] GeeksforGeeks. Available at: ¡https://www.geeksforgeeks.org/twitter-sentiment-analysis-using-python/¿ [Accessed 19 Nov 2017]. .

[2] Rouse, M., 2019. Clickstream Analysis - Definition From Whatis.Om. [online] SearchCustomerExperience. Available at: ¡https://searchcustomerexperience.techtarget.com/definition/clickstream-analysis-clickstream-analytics¿ [Accessed 24 June 2019].

[3] mksaad, V., 2018. Sentiment Analysis In Arabic Tweets With Python. [online] — Motaz Saad. Available at: ¡https://mksaad.wordpress.com/2018/12/07/sentiment-analysis-in-arabic-tweets-with-python/¿ [Accessed 13 nov 2018].

[4] G. Chowdhury. "Natural language processing," Annual review of information science and technology, vol. 37, pp. 51-89, 2003.

[5] Abbas Q. *S*emi-semantic part of speech annotation and evaluation. In: Proceedings of ACL 8th Linguistic Annotation Workshop held in conjunction with COLING, Association of Computational Linguistics, 2014

[6] Raymond G. Gordon Jr. *E*thnologue: Languages of the World, Fifteenth edition. Dallas, TX.: SIL International, 2005

[7] Asif Ekbal, RejwanulHaque, Amitava Das, VenkateswarluPoka and Sivaji Bandyopadhyay. *L*anguage Independent Named Entity Recognition in Indian Languages. In Proceedings of the IJCNLP-08 Workshop on NER for South and South East Asian Languages, Hyderabad, India. Asian Federation of Natural Language Processing, 2008.

[8] E., Keyomama, T. and Paliath, V., 2014. Explain Markov-Chain Algorithm In Layman's Terms. [online] Stack Overflow. Available at: ¡https://stackoverflow.com/questions/4081662/explain-markov-chain-algorithm-in-laymans-terms¿ [Accessed 23 Jan 2015].