TEXT TOOL

Dissertation submitted to

Shri Ramdeobaba College of Engineering & Management, Nagpur

in partial fulfillment of requirement of for the award of

degree of

Bachelor of Engineering

In

COMPUTER SCIENCE & ENGINEERING

By

Nikita Godhwani [BE12CSU011]
Saumya Paigwar [BE12CSU017]
Snehal Sambare [BE12CSU022]
Vedangi Soman [BE12CSU024]
Wani Bisen [BE12CSU033]

Guide

Prof.(Mrs.) A. M. Karandikar

Computer Science & Engineering
Shri Ramdeobaba College of Engineering & Management, Nagpur 440013
(An Autonomous Institute affiliated to Rashtrasant Tukdoji Maharaj Nagpur University)

April – 2016

TEXT TOOL

Dissertation submitted to

ShriRamdeobaba College of Engineering & Management, Nagpur

in partial fulfillment and requirement of for the award of

degree of

Bachelor of Engineering

In

COMPUTER SCIENCE & ENGINEERING

By

Nikita Godhwani [BE12CSU011]
Saumya Paigwar [BE12CSU017]
Snehal Sambare [BE12CSU022]
Vedangi Soman [BE12CSU024]
Wani Bisen [BE12CSU033]

Guide

Prof.(Mrs.) A. M. Karandikar

Computer Science & Engineering
Shri Ramdeobaba College of Engineering & Management, Nagpur 440013
(An Autonomous Institute affiliated to Rashtrasant Tukdoji Maharaj Nagpur University)

April 2016

SHRI RAMDEOBABA COLLEGE OF ENGINEERING & MANAGEMENT

(An Autonomous Institute affiliated to Rashtrasant Tukdoji Maharaj Nagpur University)

Department of Computer Science Engineering

CERTIFICATE

This is to certify that the thesis on **"Text Tool"** is a bonafide work of Nikita Godhwani, Saumya Paigwar, Snehal Sambare, Vedangi Soman, Wani Bisen submitted to the RashtrasantTukdojiMaharaj Nagpur University, Nagpur in partial fulfillment of the award of a Bachelor of Engineering, in Computer Science has been carried out at the Department of Computer Science & Engineering, ShriRamdebaba College of Engineering & Management, Nagpur during the academic year 2015-2016

Date:

Place:

Prof. (Mrs.) A. M. Karandikar	Dr. M. B. Chandak
Project Guide	H. O. D.
Department of Computer Science	Department of Computer Science
& Engineering	& Engineering

Dr. R. S. Pande

Principal

DECLARATION

We, hereby declare that the thesis titled **"Text Tool"** submitted herein has been carried out in the Department of Computer Science & Engineering of Shri Ramdeobaba College of Engineering & Management, Nagpur. The work is original and has not been submitted earlier as a whole or part for award of any degree/ diploma at this or any other institution / University.

Date:

Place:

Group Members

_____
(Nikita Godhwani)

_____
(Saumya Paigwar)

_____
(Snehal Sambare)

_____
(Vedangi Soman)

_____
(Wani Bisen)

APPROVAL SHEET

This thesis/dissertation/report entitled **Text Tool** by Nikita Godhwani, Saumya Paigwar, Snehal Sambare, Vedangi Soman, Wani Bisen is approved for the degree of Bachelor of Computer Science & Engineering.

Name & signature of Supervisor(s)

_____

_____

Name & signature of External Examiner(s)

_____

_____

Name & signature RRC Members

_____

_____

Name & signature of HOD

_____

_____

Date:

Place:

ACKNOWLEDGEMENT

We would like to offer our sincere and well devoted thanks to Dr. M. B. Chandak, Head of Department,CSE for this continued encouragement and giving us the opportunity to work on this project.

We express our sincere thanks to Prof. A. M. Karandikar for her inputs, views and guidance all along the development of project. We owe profound gratitude for her opinion, sharing facts and add-on knowledge on the project from its inception till completion.

We also thank the staff of CSE Department for their support and help for making this project successful.

Group Members:

Nikita Godhwani [BE12CSU011]

Saumya Paigwar [BE12CSU017]

Snehal Sambare [BE12CSU022]

Vedangi Soman [BE12CSU024]

Wani Bisen [BE12CSU033]

ABSTRACT

Data present in unstructured form can be processed and analyzed to extract relevant information. This unstructured data can also be transformed into user required form. Text Processing includes ability to draw insights from data contained in unstructured materials. Text analytics refers to the extraction of useful information from text sources. This includes restating the important points of text in compressed form, assigning the most appropriate meaning to a ambiguous word within a given context, analysis of Facebook/Twitter to monitor topics that are currently popular on social networking sites , word cloud representation of text and many other text processing tasks. The objective of this application is to bring text processing tasks at a single place. This includes top n sentences finder, ambiguity resolver, current trend finder, trends plotter and word cloud.

The top n sentences software was made using Python Programming language and the Natural Language Toolkit (NLTK) provided by it. For finding top n sentences the text is split into words. The stop words and punctuation are removed. The words are ranked and then the sentences are ranked according to the component words in them. Then the sentences having highest rank are returned. This gives the most important sentences from the entered text thus saving the reader's time.

The Word Sense Disambiguation software was made using Python and the NLTK provided by it. For disambiguation of word the sentence is lemmatized and the stop words and punctuations are removed. Out of all possible senses of the ambiguous word the senses which are in database are taken for further processing. The dependencies of the left and the right words are obtained from metadata. The senses are ranked by adding up the dependencies and the sense with maximum rank value is taken as best sense for given ambiguous word.

Current trend generator was made using Python, Java and NLTK. Facebook4j was used to retrieve posts from Facebook pages and plotly was used to plot a graph giving the top 5 trends. For calculating the trends the posts are retrieved from Facebook and the URLS, stop words and punctuation marks are removed, then the relative frequency of each word is calculated. Top 5 topics are selected and are displayed on GUI. Top 5 topics are mapped on a graph and the topics are linked with Google search.

Trend plotter was made using Python, Java and NLTK. Twitter4j was used to retrieve tweets and MatPlotLib was used to plot 3D graph giving the top 5 trends for the selected cities.

Word Cloud a visual representation of text data was made using Python and NLTK. The sentences are lemmatized and the stop words and punctuations are removed. Frequency is calculated for each word and size is allocated according to the frequency.Frequency wise sorting is done on the words and their positioning is done on rectangular tag.

Keywords: Text processing, top n sentences, ambiguity resolver, current trend finder, trends plotter, word cloud, Facebook, Twitter, Frequency, grap

CONTENTS

List of Figures

Chapter 1

Introduction

1.1 Problem Definition

Large amount of data present in unstructured form can be processed and analyzed to produce information in user required form .Text processing and analysis tasks are of various kinds based on the way a text is processed and the form of output it generates. Thus any user wanting to process the text requires various software and modules to get the required output format. User may want to view important points of the text and visualize the text in word cloud format and also disambiguate the same. So text tool aims at providing user various text analysis and processing capabilities in single software.

1.2 Objectives

The proposed system aims to give various text processing tasks at a single place so that user can find top n sentences, can perform ambiguity resolution and visual representation(word cloud) of text at a single place. Also finding top trending topics and topographic news according to cities on the same system.

1.3 Technology Used

The software was developed using Python Programming Language and Natural Language Toolkit (NLTK) provided by Python and Java. Python is a widely used general purpose programming language. It is frequently used for Natural Language Processing activities. NLTK is a leading platform for building Python Programs to work with natural languages. It provides:

- Easy-to-use interfaces
- Over 50 corpora and lexical resources

Java provides way to integrate Facebook and Twitter using Facebook4j and Twitter4j. With Facebook4j, application can be easily integrated with Facebook API and with Twitter4j,application can be easily integrated with Twitter API.

Plotly was used to represent current trends in 2D form and MatPlotLib was used for 3D representation of topographic news.

1.4 Organization of report

The report is organized as follows. We first describe system architecture of top n sentences finder, ambiguity resolver, Trends finder,Trends Plotter and Word cloud consisting the flow diagrams and the GUI of all the modules followed by working of all modules and system description then future scope of the software .

Chapter 2

Review of Literature

2.1 Overview

2.1.1 Top n Sentences

This module finds out the relevance of sentences in text, by assigning ranks to the sentences based on word count and title of the article. Then the number of sentences (as required by the user) is returned in descending order of their ranks.

2.1.2 Word Sense Disambiguation (WSD)

A sentence is taken as input. The sentence is analyzed to find out the ambiguous word and its meaning is given as output

2.1.3 Trendy

The Trendy module returns the top 5 trending topics on Facebook based on selected domain

2.1.4 Trends Plotter

Trends plotter maps the top 5 trending topics of a city on a 3-D map, where peaks represent the rank of the trending topic in that city.

2.1.5 Word Cloud

A word cloud is a visual representation of text data, typically used to depict keyword metadata (tags) on websites, or to visualize free form text. Tags are usually single words, and the importance of each tag is shown with font size or color.This format is useful for quickly perceiving the most prominent terms and for locating a term alphabetically to determine its relative prominence.

2.2 History

2.2.1 Top n Sentences

The history of automatic i.e. computerized summarization began 50 years ago. Luhn's method uses term frequencies to appraise eligibility of sentences for the summary. The next remarkable progress happened ten years later. Edmundson's work introduced hypothesis concerning high information value of title phrases, sentences from the beginning and from the conclusion of the article, sentences containing cue words and phrases as "important", "results are", "paper introduces", etc. Sentence revision was historically the first language generation task attempted in the context of summarization. It involves re-using text collected from the input to the summarizer, but parts of the final summary are automatically modified by substituting some expressions with other more appropriate expressions, given the context of the new summary.

2.2.2 WSD

WSD is one of the oldest problems in computational linguistics. It was in the 1940s that WSD was first formulated as a separate task by Weaver. It was acknowledged from the very beginning that the task is crucial and non-trivial. The 1980s witnessed a turning point in WSD research, and large scale corpora and other lexical resources became available. Before the 1980s much of WSD research depended on handcrafting of rules. Now it became possible to use knowledge extracted automatically from the resources.

Lesk came up with a simple yet seminal algorithm that used dictionary definitions from the Oxford Advanced Learner's Dictionary (OALD), and this marked the beginning of dictionary-based WSD. WordNet made it possible for all the researchers to have easy and free access to a standardized inventory using which to compare their work. Its hierarchical structure, synsets, and other such features made it the most used general sense inventory in WSD research. The mainstream approach in WSD so far has been supervised learning, where systems are trained on manually tagged corpora. In SensEval-3 (2004), a conclusion was reached that WSD in itself has reached a performance plateau, and no significant leap in the results obtained already is possible.

2.2.3 Trendy

Twitter's data visualization scientist Nicolas Belmonte proposed to represent thousands of tweets cropping up at various places in the world as a Topographical map. This map would consist of high peaks and low valleys, where high peaks represent highest activity. They also represent popular and widely repeated content reflecting the significant role these topics play in twitter discussions.

In June 2013, Twitter showed a 2-D map showing every geotagged tweet since 2009. Later on this map was converted to a 3-D representation to enable better visualization.

2.2.4 Trends Plotter

The Pew Research Centre is creating intense and visually rich maps or charts to help understand twitter conversations. The team has collected, analyzed and visualized social media network data using NodeXL.

NodeXL is a general purpose network analysis application that supports network overview, discovery and exploration. By analyzing many thousands of twitter conversations they have identified various structures of twitter conversation networks. These structures tell a story about the nature of conversation.

2.2.5 Word Cloud

Word cloud generators has been implemented by various people, some of the word cloud generators are wordle, Jason Davis's word cloud, Amueller's word cloud(on Github).

The word cloud has been generated on the basis of the frequency of words.

Chapter 3

Proposed Approach and System Architecture

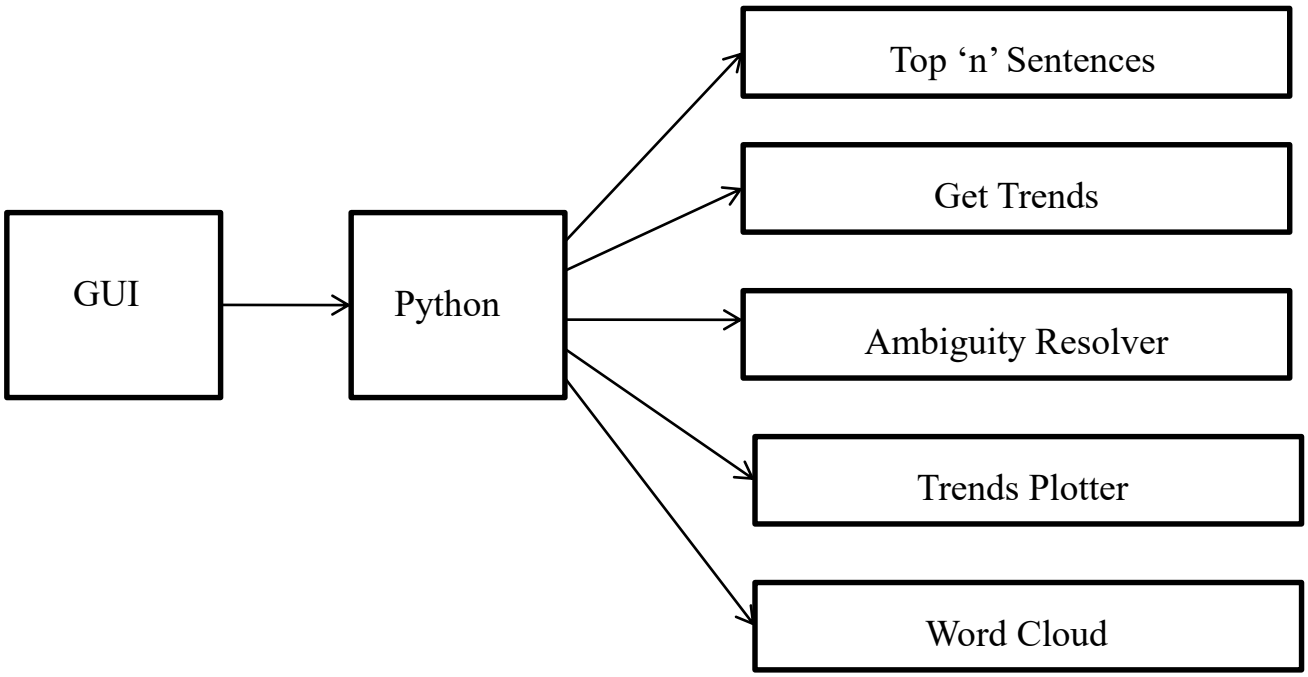

Fig. 3.1: System Architecture

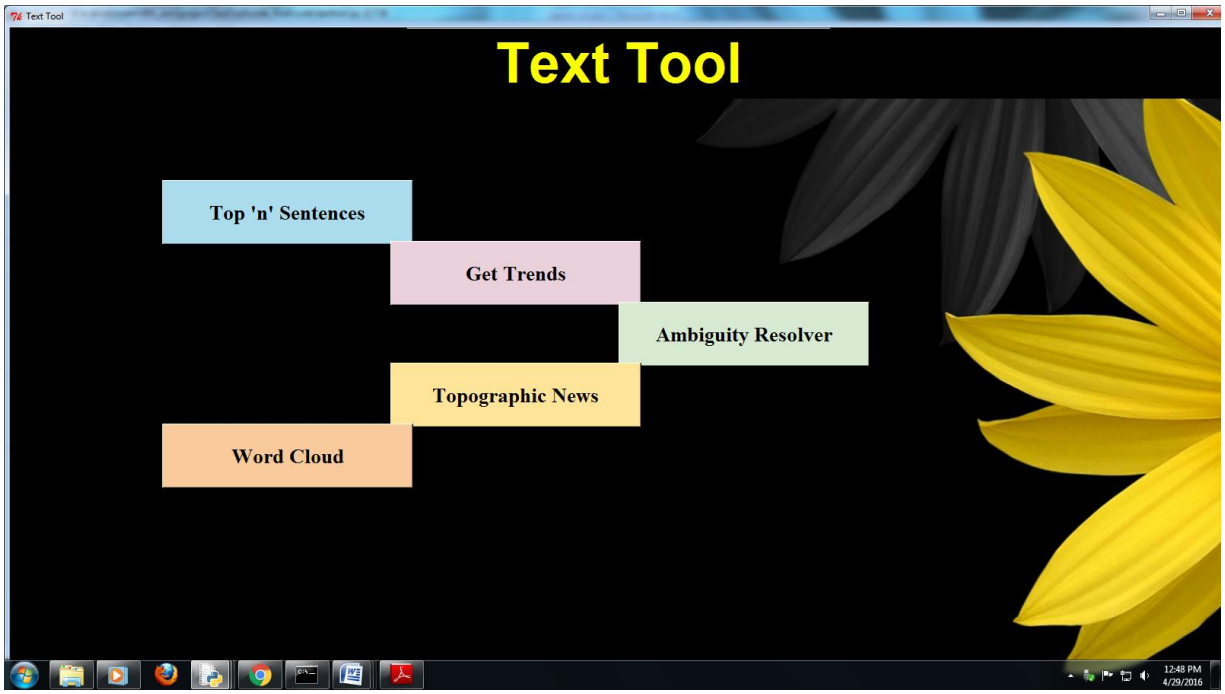

Fig. 3.2: GUI for Text Tool

In this project the major part of processing of textual data has been done using Natural language toolkit (NLTK). The retrieval of data has been done using Facebook4j, Twitter4j. Graph plotting has been done using Matplotlib library. The GUI of text tool has buttons on it for five modules which are shown in figure(Fig). Any one module can be run by clicking the button on GUI. The modules are described below

3.1 Top' n' Sentences

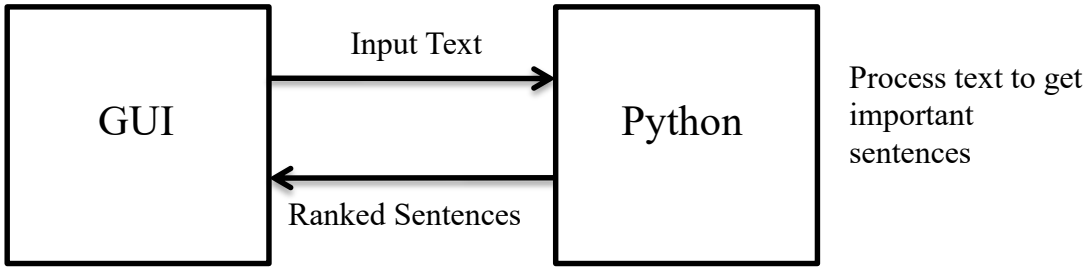

Fig. 3.3: System Architecture for top n sentences

The program takes text as input either from a file or directly from the GUI, performs processing tasks and returns to the GUI output as Top Ranked 'n' Sentences.

Fig. 3.4: GUI for Getting Ranked Sentences

3.2 WSD

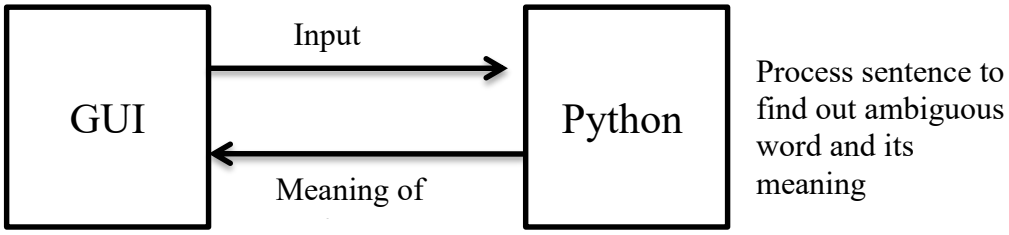

Fig. 3.5: System Architecture for WSD

The program takes a sentence as input, applies processing and finds the ambiguous word and its correct meaning in the given context and returns this as output to the GUI.

Fig. 3.6: GUI for Ambiguity Resolver

3.3 Trendy

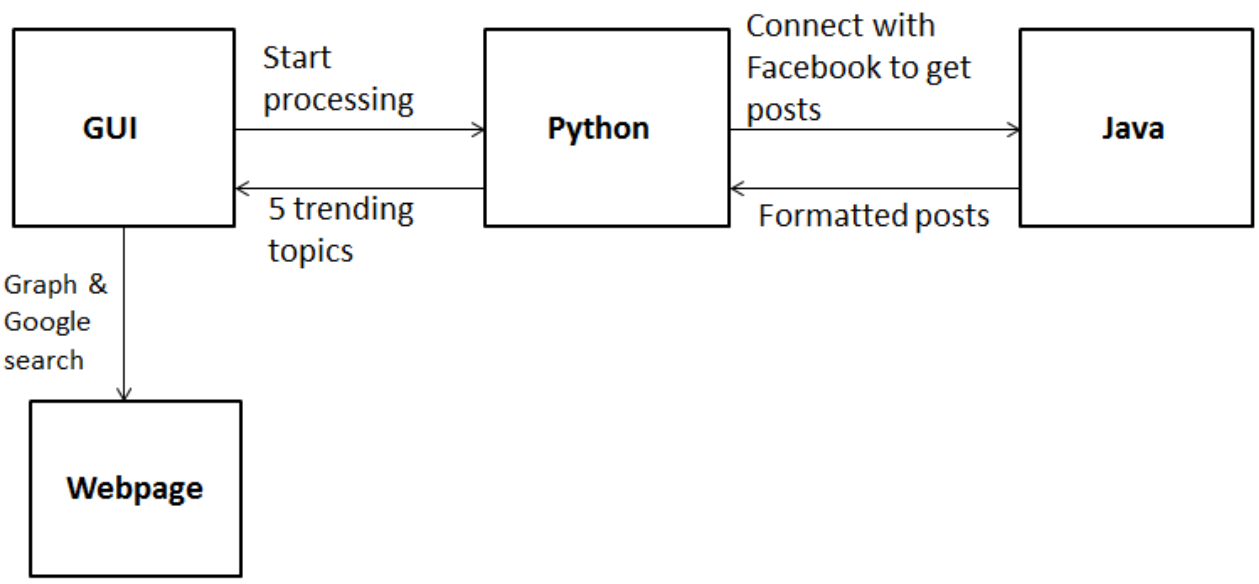

Fig. 3.7: System Architecture for Trendy

Trend Plotter obtains posts from domain specific Facebook pages using Faecbook4j, finds out trending topics from obtained posts and returns them as output. These trending topics are also linked to Google search to facilitate better knowledge and understanding of the topics

Fig. 3.8: GUI for Trend Getter

3.4 Trends Plotter

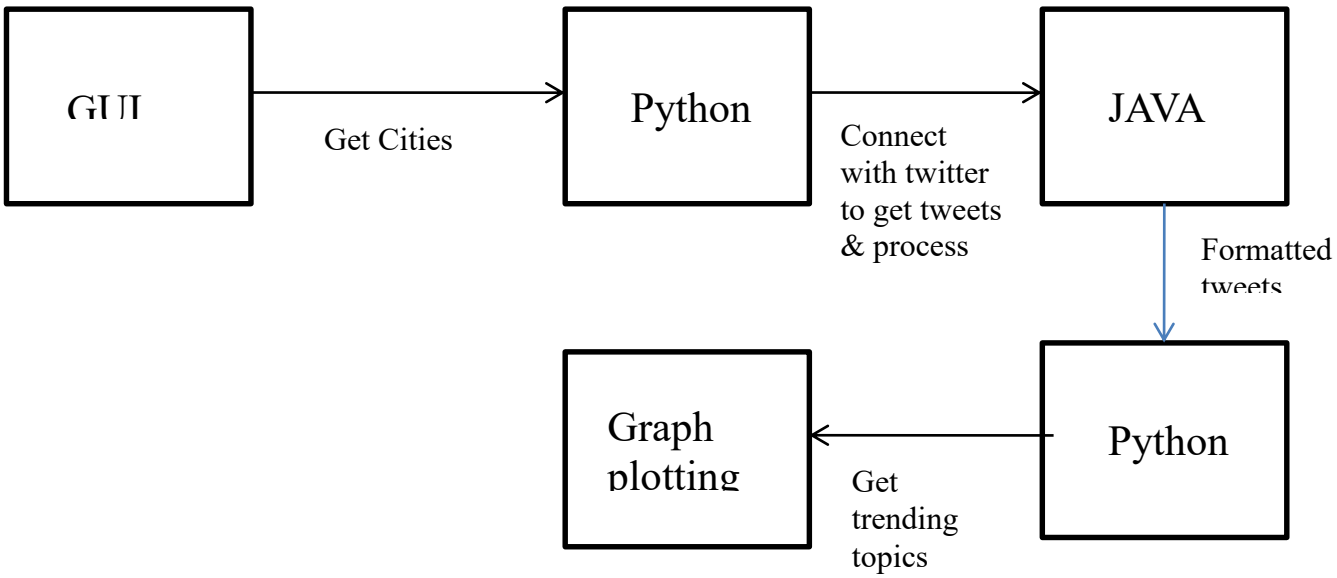

Fig. 3.9: System Architecture for Trends Plotter

On clicking the "Trends Plotter" button on Text Tool's GUI, another GUI for trends plotter is opened as shown in figure Fig 3.2

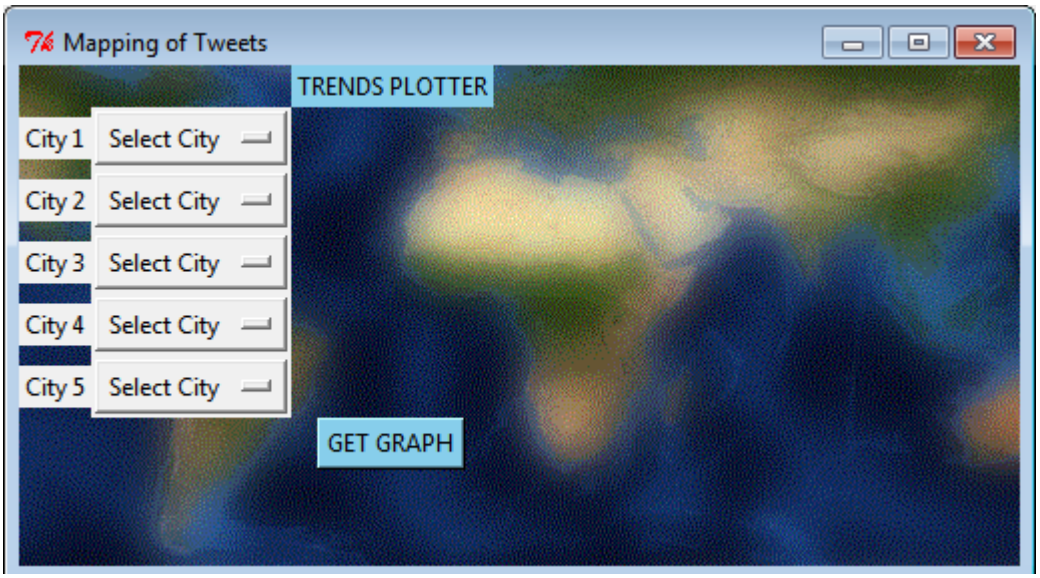

Fig. 3.10: GUI for Trends Plotter

The cities are selected and on clicking the get graph button the processing starts. From python module the java code is called which connects with twitter and retrieve tweets of a city. The tweets are formatted to be processed and send to python module to get the top 5 trending topics of a city. These topics are then plotted on a 3D graph. For each city this process continues and graph is plotted. Finally after processing for all the cities provided by the user the 3D graph is displayed.

3.5 Word cloud

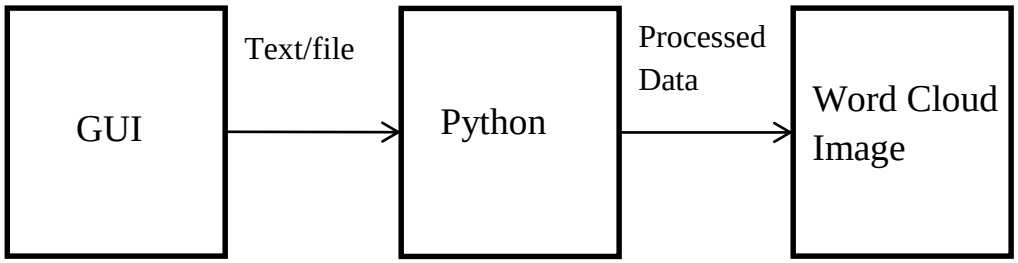

Fig. 3.11: System Architecture for Word cloud

On clicking the word cloud button on main GUI another GUI window pops up as shown in figure fig 3.12. The input to the word cloud module can be given in two ways first as an input file and second as text written manually or copied in the text area provided on word cloud GUI.

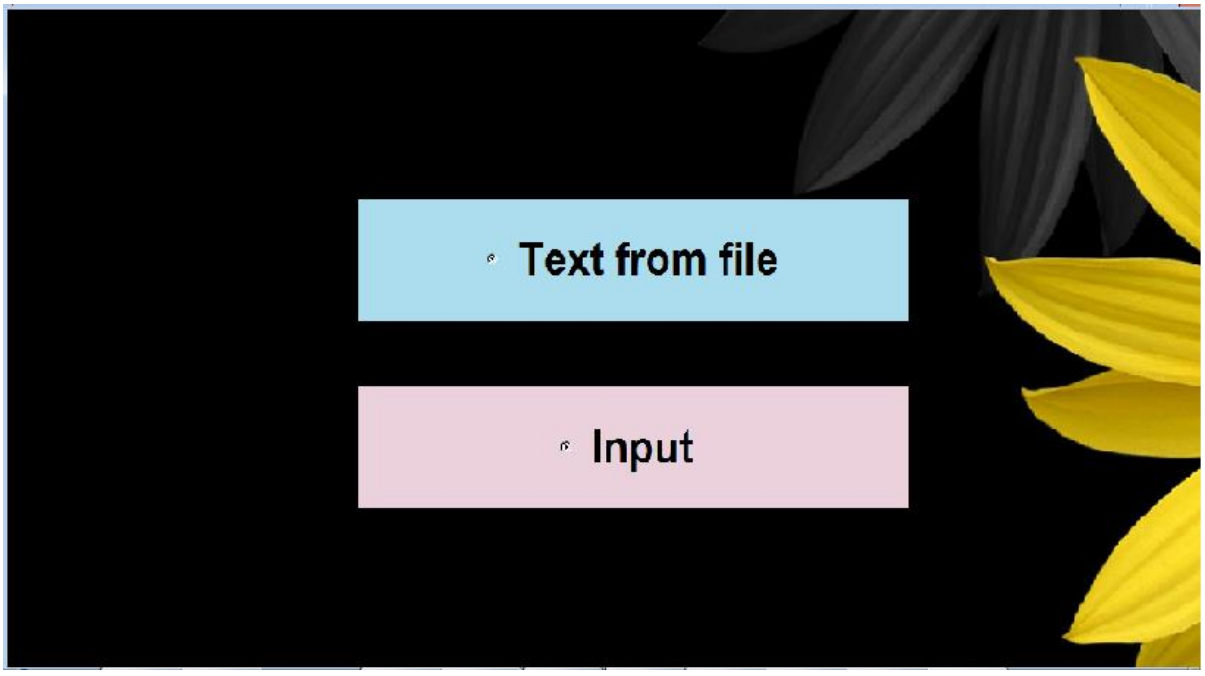

Fig. 3.12: GUI for Word cloud

On clicking the get cloud button the text is processed and the word cloud image is displayed as shown in figure 5.10.

Chapter 4

Modules Used

Nltk.tokenize

This module is used for splitting any string into substrings according to specified conditions. From this module the method ***word_tokenize*** and ***sent_tokenize*** were used. Sent_tokenize divides the paragraphs into sentences. Word_tokenize divides the sentence into words.

Nltk.corpus

This module has a set of large bodies of linguistic data (corpora). It defines a collection of corpus reader classes which can be used to access these corpora. Stopwords are most common words such as the, is, on, at, etc. The method call ***stopwords.words ('english')*** was used to remove these unimportant words. ***WordNet*** corpus was used to find out senses of ambiguous words. The method used is ***WordNet.synsets ('ambiguous word')***

Collections

The collections module provides container datatypes. It implements specialized container datatypes which provide an alternative to Python‟s built-in containers. The method ***defaultdict(a)*** from the collections module is used to declare a variable of type 'a'.

String.punctuation

It is a string of ASCII character which is considered as punctuations in the c language.

Heapq

This module provides an implementation of the heap queue (priority queue) algorithm. Any populated list can be converted to a heap by means of the heapify() function. Here the method nlargest was used to get the most important n elements and return them in GUI.

Pos tagger:

A part-of-speech tagger, or POS-tagger, processes a sequence of words, and attaches a part of speech tag to each word

Lemmatizer:

A lemmatizer is a tool from Natural Language Processing which does full morphological analysis to accurately identify the lemma for each word.Lemma is the base or dictionary form of a word. From Nltk.stem.wordnetthe method wordnetLemmatizer() has been used. This module lemmatizes usingWordNet's built-in morphy function. Returns the input word unchanged if it cannot be found in WordNet.Jpype

Since python doesn‟t support Facebook4j and Twitter4j, Jpype was used to bridge python project with java libraries.

Facebook4j

Facebook4j is an unofficial java library for the Facebook graph API.

Following methods of Facebook4j have been used

setOAuthAccessToken

This method is used to set access token.

AccessToken

It is an opaque string that identifies an user, app, or page and can be used by the app to make graph API calls. When someone connects with an app using Facebook login the app will be able to obtain an access token which provides temporary secure access to Facebook APIs.

getFeed(String id):

It returns a page‟s wall.

Id- the ID of a page

The feeds which are returned are collected in ResponseList

ResponseList:

Is a list of names, address and other customer data compiled by company based on sign-ups subscriptions and other commercial transactions.

Twitter4j:

Twitter4J is an unofficial Java library for the Twitter API. With Twitter4J, you can easily integrate your Java application with the Twitter service. Twitter4J is an unofficial library.

Following methods of twitter4j have been used

setOAuthConsumerKey("token")

setOAuthConsumerSecret("token")

setOAuthAccessToken("token")

setOAuthAccessTokenSecret("token")

These methods are used for setting up tokens to establish connection with twitter.

Search(query): This method is used to search the tweets with the query given in ‘query’ field.

GeoCode(new GeoLocation(lat,lon),res, resUnit): this method is used to get tweets of particular region by providing latitude and longitude , range of area upto which tweets are to extracted in ‘res’ field and unit of range (miles/kilometers/meters etc) in ‘resUnit’ field.

getText(): to get the text associated with the tweet.

List<status>: used to store the tweets retrieved.

Tkinter

Tkinter is the python interface to Tk module. It is dominantly used for GUI programming in Python i.e. even though other modules for GUI programming are available in Python Tkinter is the most commonly used. Labels, buttons, textboxes, etc., as well as the GUI window was developed with the help of Tkinter.

PIL

PIL module is used for inserting graphics in the GUI screen.

Matplotlib:

Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy.

System Description

4.1 Top ‘n’ Sentences

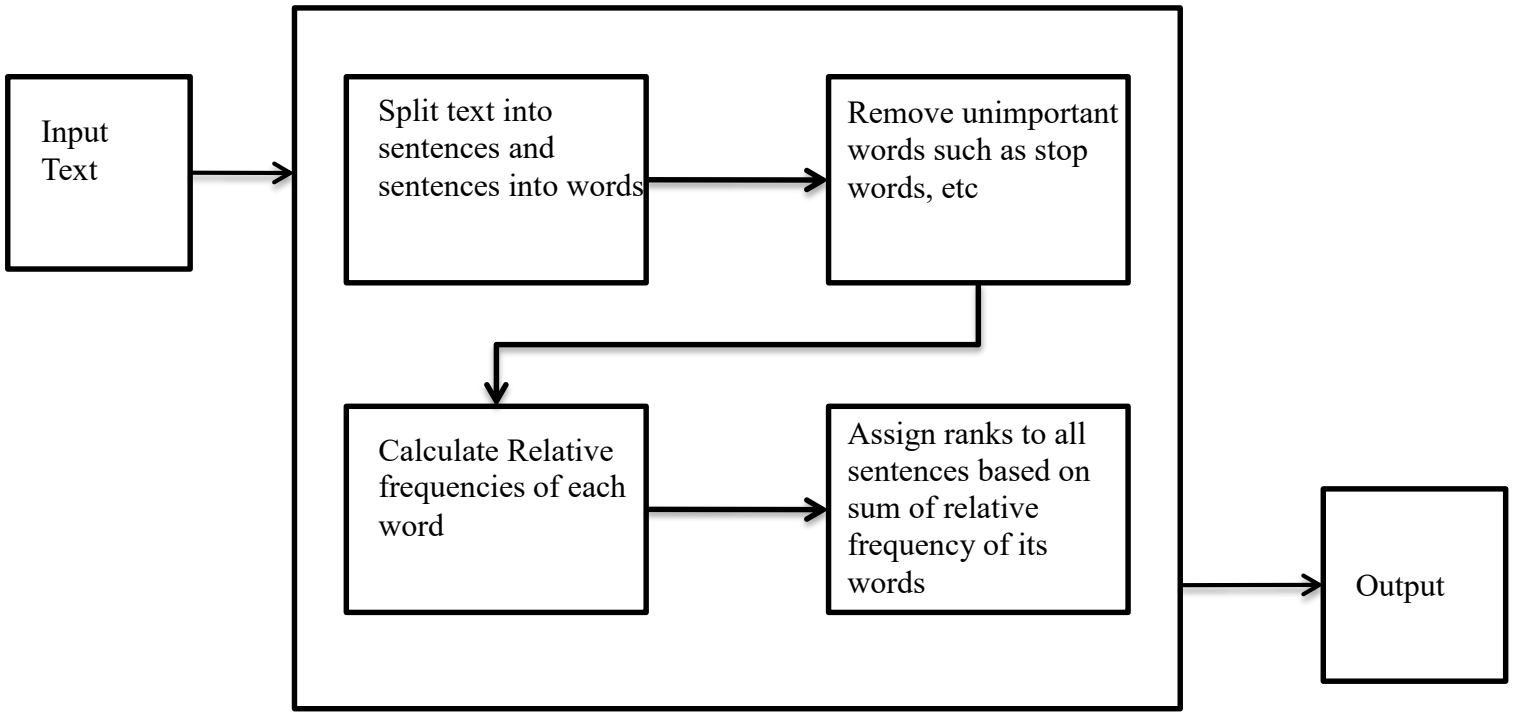

Fig. 4.1: Flow Diagram of Top ‘n’ Sentences

Steps:

1) The news article and the no. of statements required in the summary are entered.
2) The entered text is split into sentences. These sentences are then split into words.
3) These words are filtered by removing stop words and punctuations.
4) The frequency of each word (from the remaining words) is calculated.
5) The frequency of each word relative to the word having highest frequency is then calculated. The words with relative frequencies within the range 0.1 – 0.9 only are considered, so as to ignore the unimportant words.
6) The rank of each statement in the input text is calculated by adding up the relative frequencies of the words appearing in those statements.
7) These sentences are then sorted using nlargest method of heapq which returns ‘n’ sentences having highest ranks.
8) These sentences are then returned as summary statements.

4.2 WSD

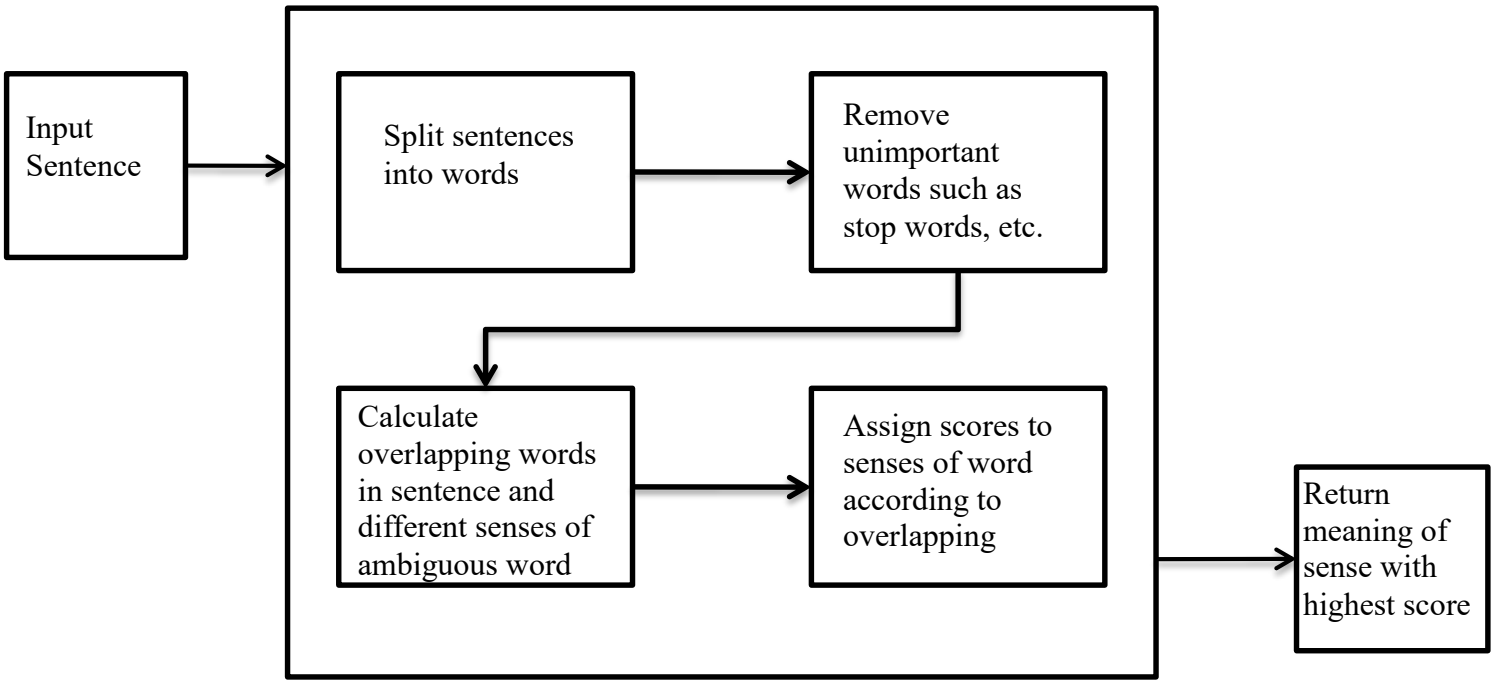

Fig. 4.2: Flow Diagram of WSD

Steps:

1) A sentence is entered to the GUI
2) This sentence is split into words
3) The best_sense for the ambiguous word is initially set to the most frequently used sense of the word and max_overlap is initialized to 0
4) For each of the possible senses for the ambiguous word
    - Compute overlap is calculated as number of words overlapping in the sentence and the meaning of 'this sense'
    - If overlap >max_overlap then max_overlap is set to overlap and best_sense is set to 'this sense'
5) The meaning of best_sense is returned as output

4.3 Trendy

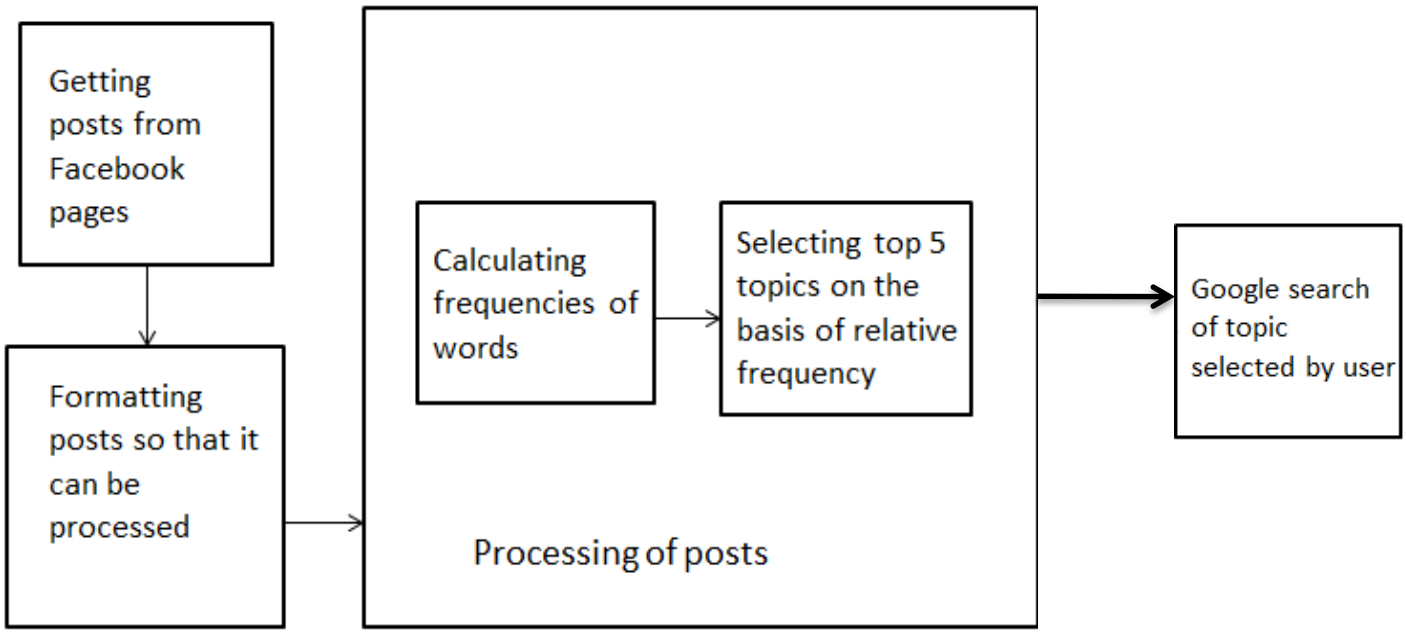

Fig. 4.3: Flow Diagram of Trendy

Steps:

1) Using access token posts from various Facebook news pages were retrieved using Facebook4j(a wrapper for the Facebook API), which were stored in a text file.
2) This file was given as input to a java code to remove http links.
3) The lines containing http links were split on the basis of spaces and the strings containing http links were excluded and remaining strings were written to a file.
4) This file was read as a string which was in ASCII format, the string was converted to UNICODE format
5) This formatted string was tokenized into sentences and each sentence thus obtained was split into words
6) The frequency of each word excluding, “stop words” (words like this, they, etc.) was counted
7) The relative frequency of each word was calculated by dividing its frequency by the highest frequency. The words with relative frequencies within the range 0.1 – 0.9 were arranged in decreasing order. This range ensures that words that either appear too many times(relative frequency > 0.9) or occur very infrequently(relative frequency < 0.1) are ignored since they are of least importance
8) The top 5 words thus obtained were returned as trending topics
9) These topics were also linked with Google Search

4.4 Trends plotter

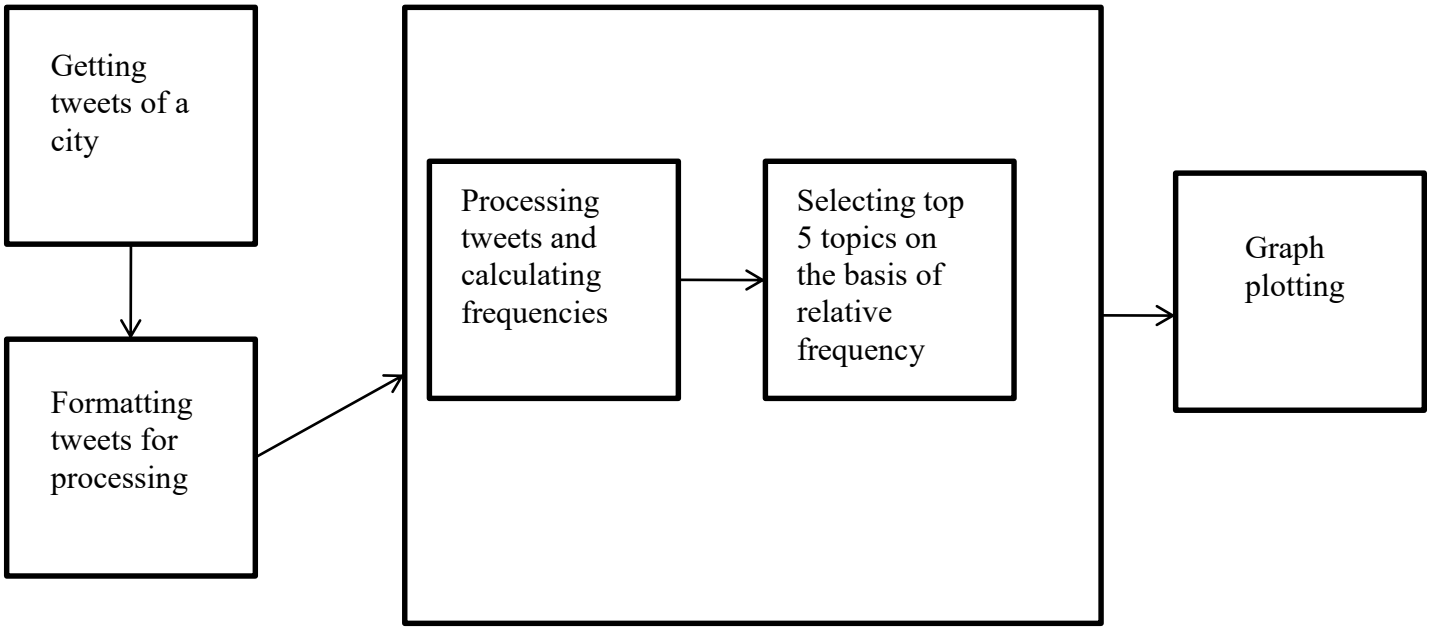

Fig. 4.4: Flow Diagram of Trends plotter

4.4.1 Retrieval and processing of posts

1) Twitter4j is used to connect with twitter using ConsumerKey, ConsumerSecret, AccessToken, AccessTokenSecret.

2) A java class has been created which stores the names of ten cities (Ahmedabad, Aurangabad, Banglore, Bhopal, Delhi, Jaipur, Lucknow, Mumbai, Nagpur, Pune) and their respective latitude and longitude.

3) The tweets of a particular city within a range of 50 miles are retrieved by using latitude and longitude and stored in text file.

4) The file is given as input to java code to remove http links and Hindi text.

The lines containing http links were split on the basis of spaces and the strings containing http links were excluded and the Hindi text are removed on the basis of ASCII values and remaining strings were written to a file.

5) This file is read as a string which is in ASCII format, the string is converted to UNICODE format.

6) The tweets in this file are now lemmatized and pos tagged to get only proper noun(s) or nouns and this data is now written in a file

7) The formatted data is tokenized into sentences and each sentence thus obtained is split into words

8) The frequency of each word excluding stop words (words like this, they, etc.) is counted

9) The relative frequency of each word is calculated by dividing its frequency by the highest frequency

10) The words with relative frequencies within the range 0.1 – 0.9 are arranged in decreasing order. This range ensures that words that either appear too many times (relative frequency > 0.9) or occur very infrequently (relative frequency < 0.1) are ignored since they are of least importance

11) The top 5 words thus obtained are returned as trending topics

4.4.2 Plotting of graph

1) The trending topics obtained for a city are plotted on the graph. The above procedure is repeated for each city and trending topics are plotted on the same graph.
2) The X axis of graph represents the cities in the order entered by user through GUI.
3) The y axis represents the trending topics of all the cities given by user arranged in alphabetical order.
4) The Z axis represents the no of trending topics, which are 5. So in graph the one with the highest peak is most trending.
5) Here matplotlib uses CSV (comma separated values) file to plot the graph. For each city this file is overwritten.

4.5 Word cloud

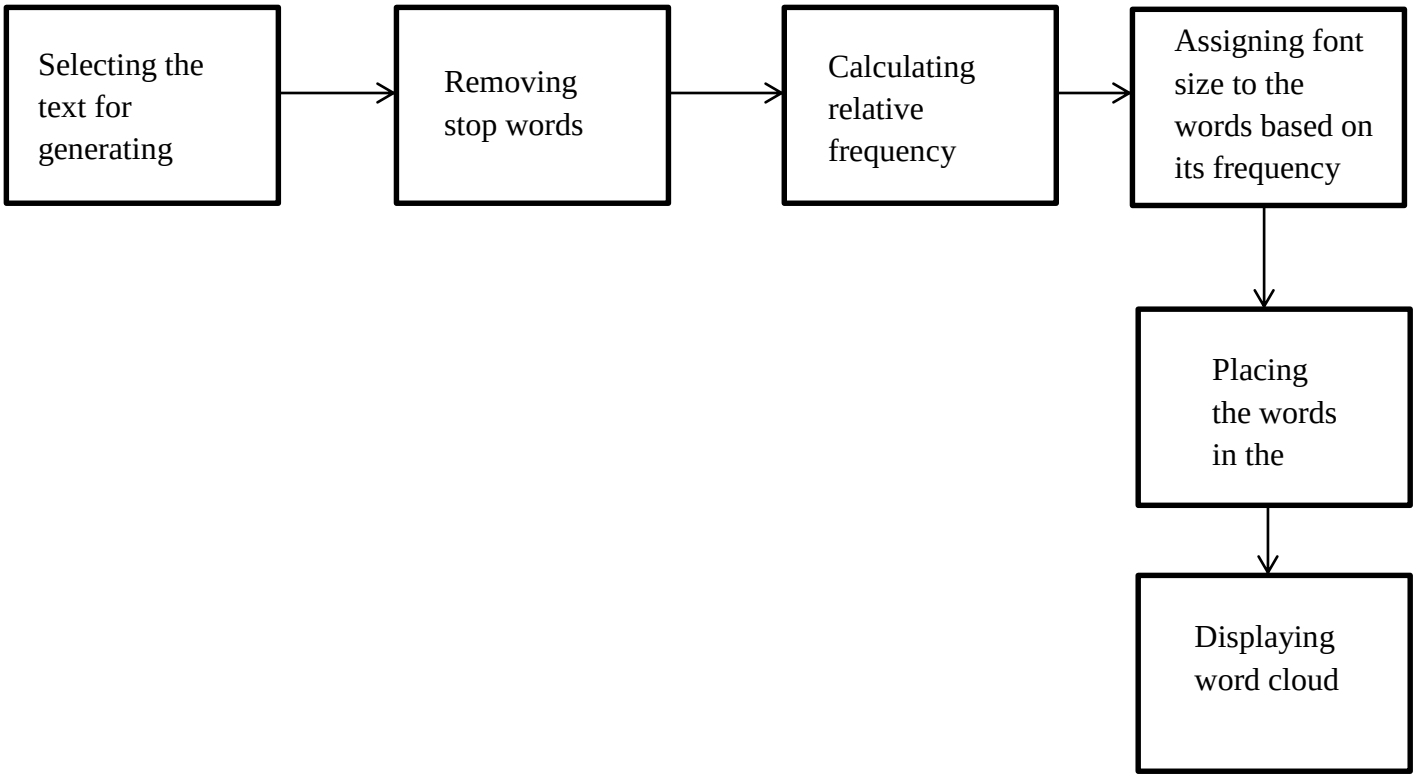

Fig. 4.5: Flow Diagram of Word Cloud

Steps

1. The text is tokenized into sentences
2. The sentences are tokenized into words
3. The stop words are removed
4. The relative frequency of each word is calculated
5. The words are sorted in descending order according to their frequencies
6. The font size is assigned to each word based on its weight(frequency)
7. The words are placed in the canvas (using integral image query) and checked if it intersects any previously placed words and the same is repeated for remaining words.

The integral image is basically a 2d cumulative sum and can be computed as

$$integral\_image = np.cumsum(np.cumsum(image, axis = 0), axis = 1)$$

Equation 4.5.1

This can be done once, and then we can look up rectangles of any size very fast. If we are interested in windows of size (w, h) we can find the sum over all possible windows of this size via

$$Area = (integral\_image[w:, h:] + integral\_image[:w, :h] - integral\_image[w:, :h] - integral\_image[:w, h:])$$

Equation 4.5.2

Chapter 5

Results and Discussion

5.1 Top ‘n’ sentences

A news article given as input

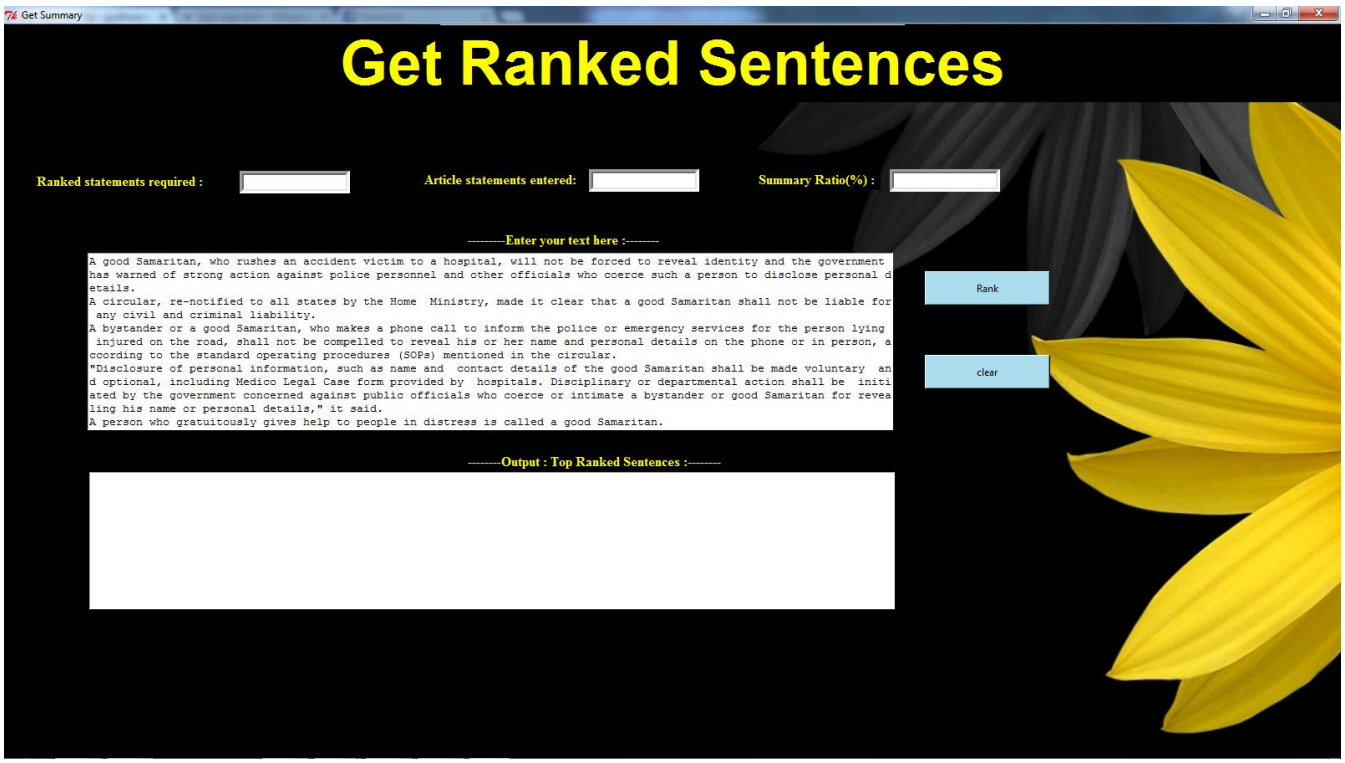

Fig. 5.1: GUI for input in Top ‘n’ Sentences

On clicking “Rank” yields following output:

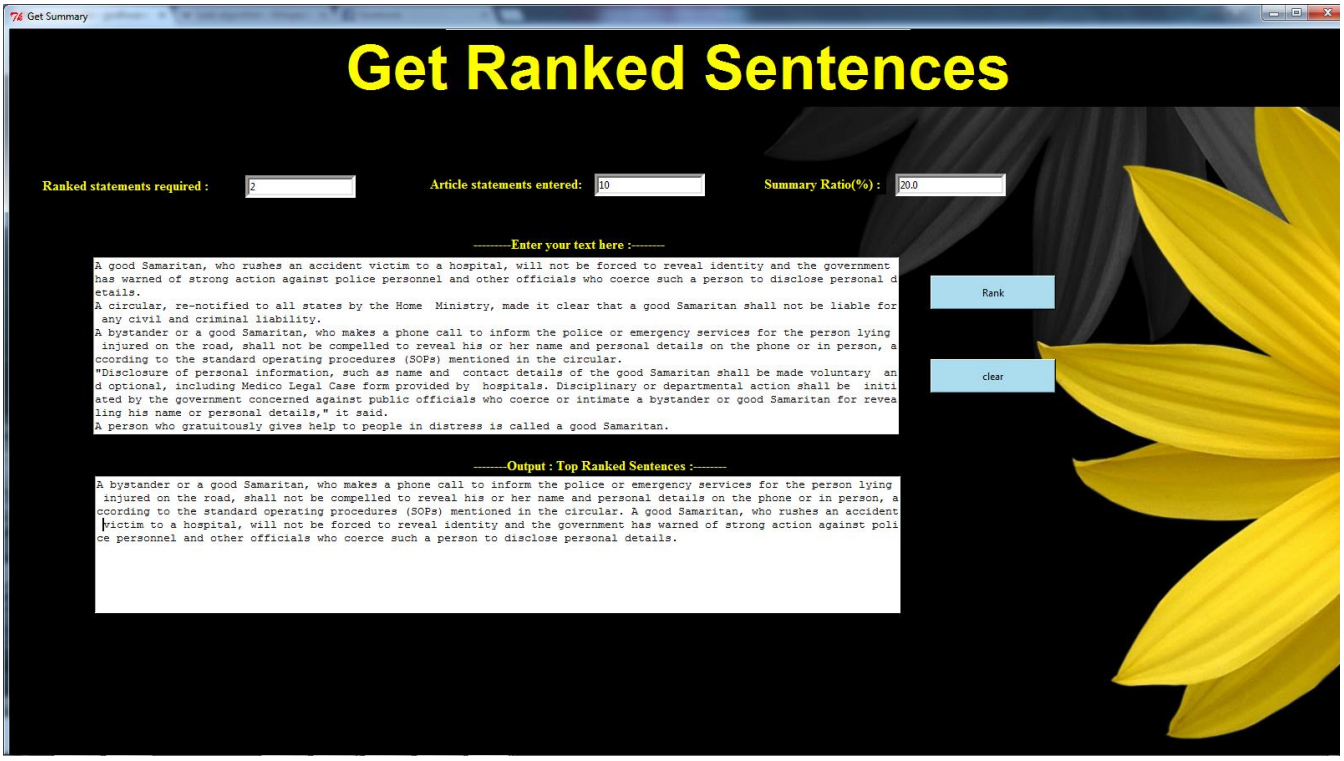

Fig. 5.2: GUI for output in Top ‘n’ Sentences

5.2 WSD

Input sentence

Fig. 5.3: GUI for input in WSD

Corresponding output (obtained by clicking “Resolve” button) meaning of ambiguous word in the sentence

Fig. 5.4: GUI for output in WSD

5.3 Trendy

The trending topics as output after running the module on 29 April 2016:

On clicking the “Get Trends Button” the screen for selection of domain appears.

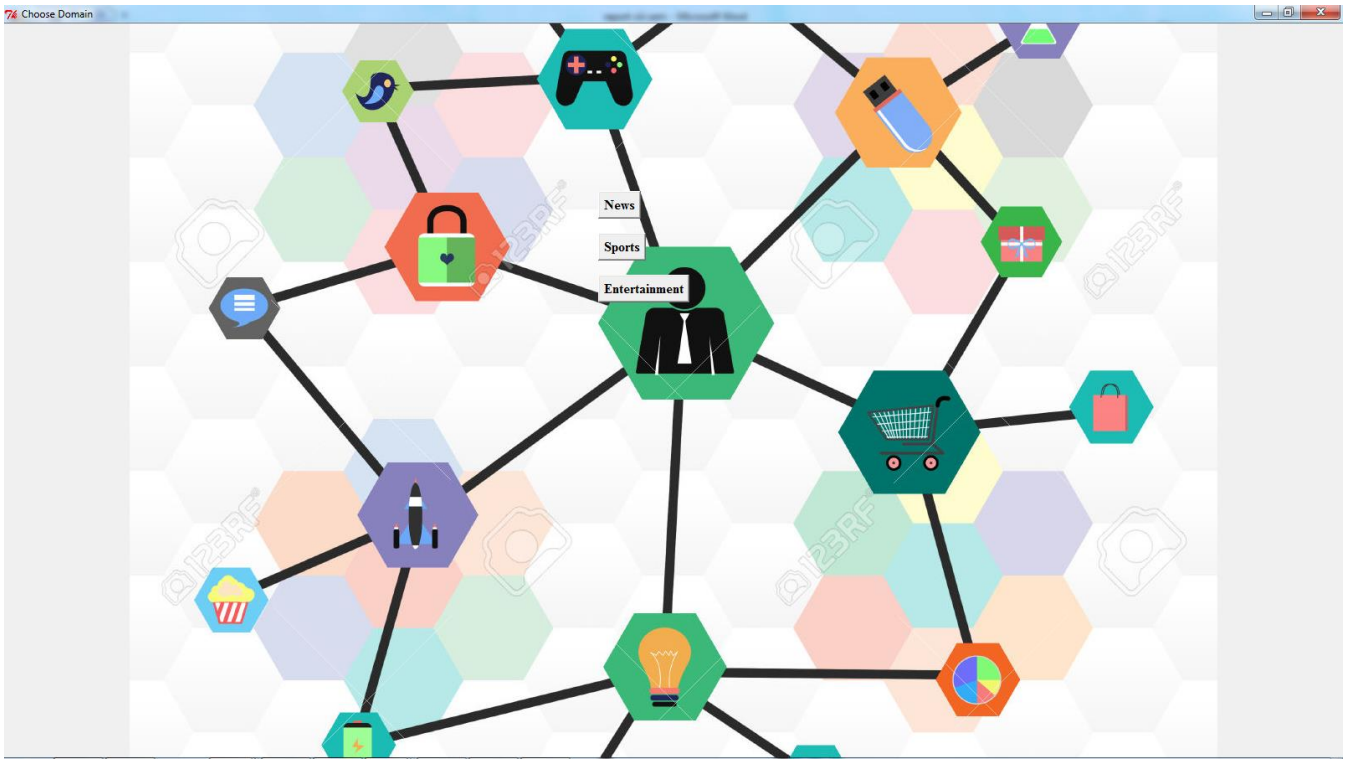

Fig. 5.5: GUI for input in Trendy

. On clicking the “News” domain button, following results are obtained:

Fig. 5.6: GUI for output in Trendy

5.4 Trends Plotter

The following are the results when the module was run on 28 April 2016

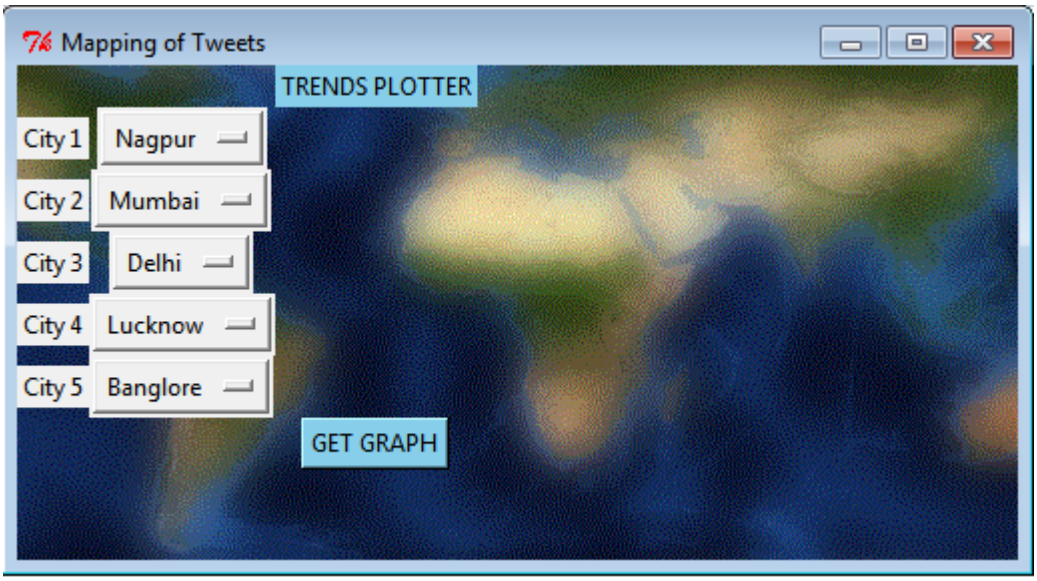

Fig. 5.7: GUI for input in Trends Plotter

On entering the following data and clicking the “Get Graph” button following graph is obtained

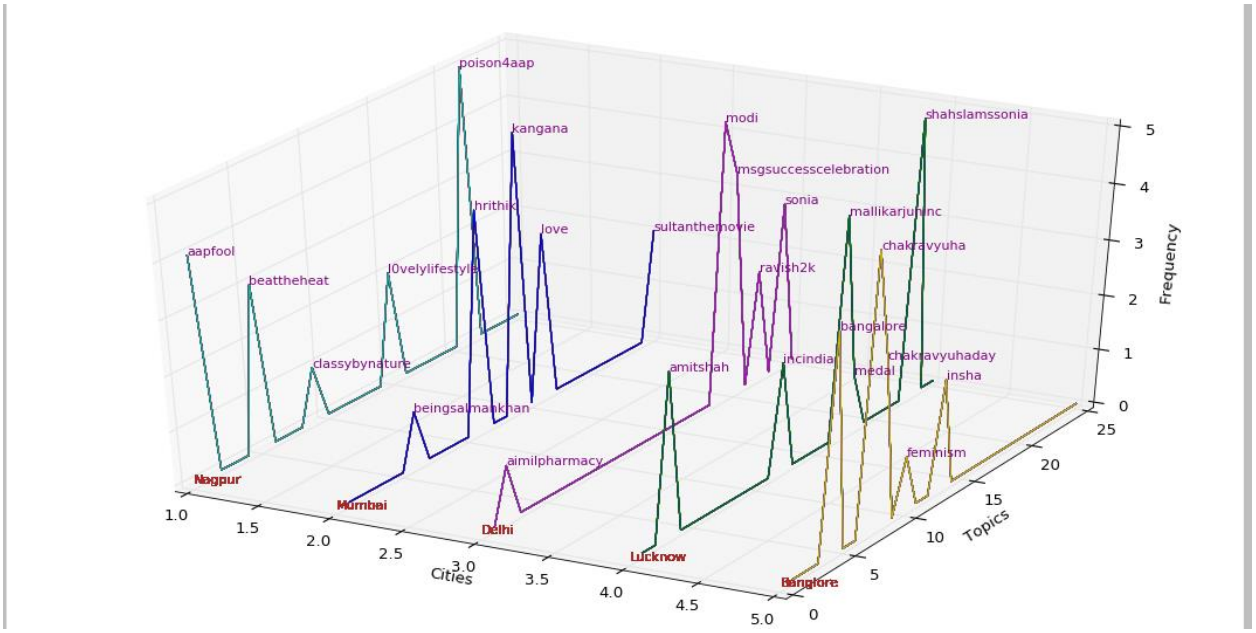

Fig. 5.8: GUI for output in Trends Plotter

The trending topics that are obtained are as follows

Nagpur : poison4aap, aapfool, beattheheat, l0velylifestyle, classybynature

Mumbia: kangana, hrithik, love, sultanthemovie, beingsalmankhan

Delhi :modi, msgsuccesscelebration, sonia, ravish2k, aimilpharmacy

Lucknow :shahslamssonia, mallikarjuninc, amitshah, incindia, medal

Banglore :chakravyuha, bangalore, chakravyuhaday, insha, feminism

5.5 Word Cloud

The word cloud generated for given input file is as shown in the figure

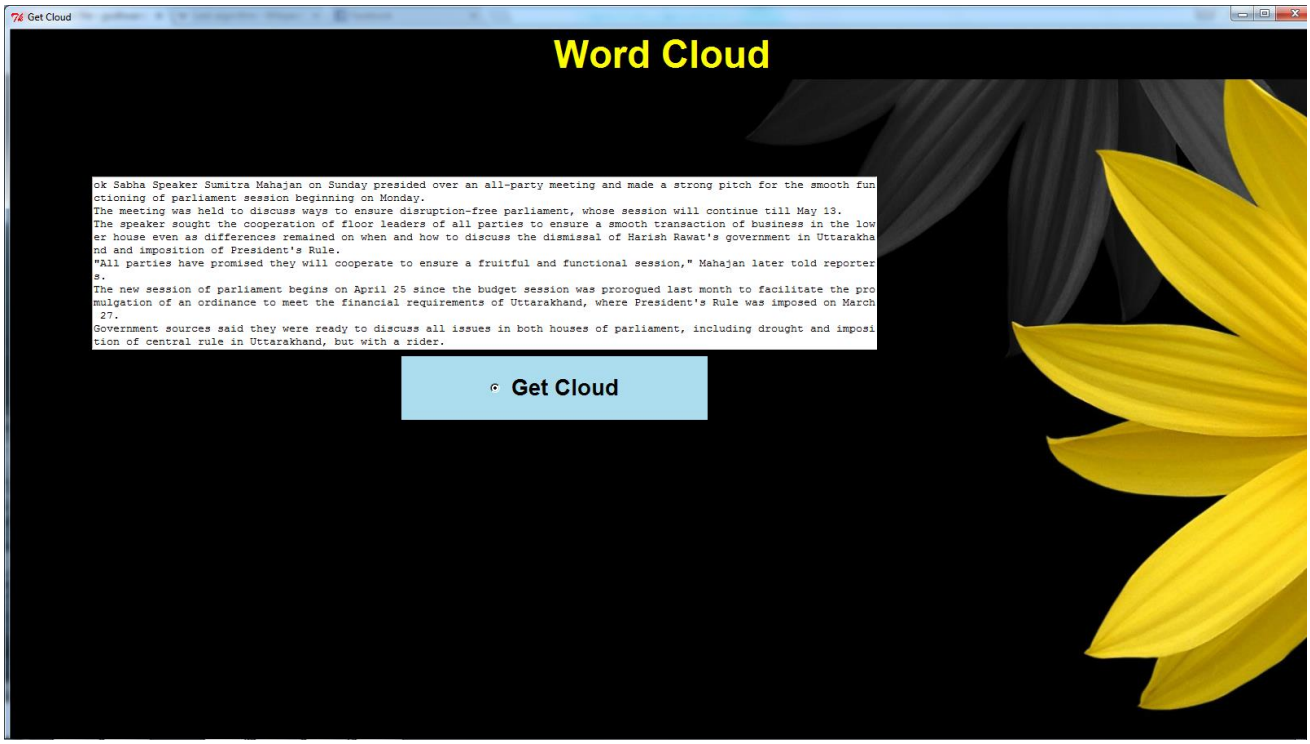

Fig. 5.9: GUI for input in Word Cloud

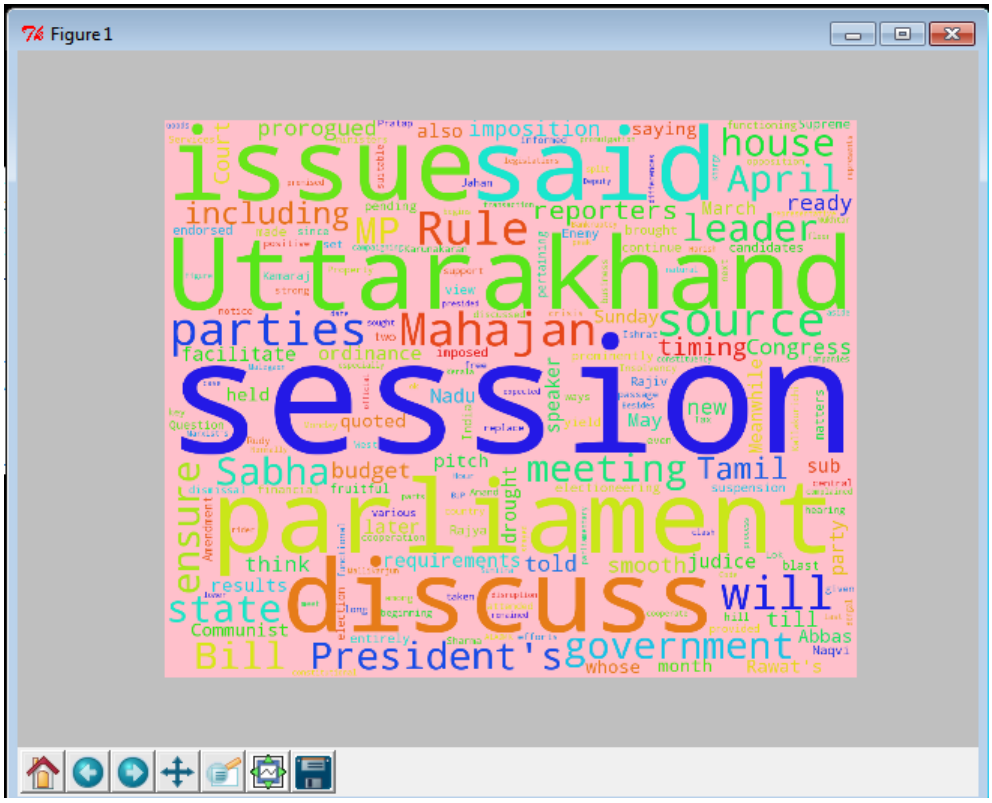

Fig. 5.10: GUI for output in Word Cloud

Chapter 6

Conclusion and Future Work

6.1 Conclusion

The ‘text tool’ has been implemented by using the five modules for text processing which involves analysis of text given by user the data from social networking sites like Facebook and Twitter. Top ‘n’ sentences have been found out by calculating the relative frequencies of words in the sentences and ranking them according to their sum value with the sentence having highest sum of relative frequencies of words in it given the first rank. In word sense disambiguation unsupervised approach has been used which gives the best sense by considering the left and right words of ambiguous word. In trendy, the process of getting trending topics from Facebook’s public pages has been explained by calculating relative frequencies of words and getting the top 5 topics on the basis of relative frequency. In trends plotter top 5 trending topics have been found out by calculating the relative frequencies of words in tweets obtained from particular region. In word cloud the relative frequency of each word has been found out and sorted in descending order and which are then displayed in word cloud image.

6.2 Future Scope

The project can be extended to add more textual processing tools. The results can be improvised by processing data syntactically and semantically, dependencies and parallel processing of modules can be implemented. Text Tool can be scaled to handle big data using big data technologies and different languages by adding dictionaries. Individual modules can be improvised. Top n sentences can be extended to implement summarization of text. Disambiguation performance can be improved further by collecting and incorporating more context knowledge. The future work includes continue building a knowledge base from different sources of text to include variety of sentences from multiple domains to enlarge scope of disambiguation. Trendy and trends plotter can be extended to include other social networking sites like “Quora”,”Google+”,”StackOverflow”. More cities and domain can be increased. Word clouds when used in websites provide navigation to visitors with instant illustration.We can represent synonymous (antonyms) words in a text using word cloud.

References

[1] AleksBuczkowski -July 2, 2013, The Topography Of Tweets – Elevation Map Of All Tweets Since 2009, Available:

[2] Liz Stinson – July 17, 2013, Geolocated Tweets Form Towering Pillars in New 3-D Maps, Available: http://www.wired.com/2013/07/what-your-tweets-look-like-as-3d-topography/

[3] Allison Stadd- July 1, 2013, The Topography Of Tweets (Interactive Model), Available: http://www.adweek.com/socialtimes/topography-of-tweets/487081

[4] Drew Desilver- February 20, 2014, How Pew Research mapped the conversations on twitter, Available: http://www.pewresearch.org/fact-tank/2014/02/20/qa-how-pew-research-mapped-the-conversations-on-twitter/

[5] Andreas Mueller –November 6, 2012, A Word Cloud in Python, Available: http://peekaboo-vision.blogspot.in/2012/11/a-wordcloud-in-python.html

[6] Suanmali, L.; Salim, N. and Binwahlan, M.S. , "Fuzzy Genetic Semantic based Text Summarization," in Ninth IEEE Dependable Autonomic and Secure Computing (DASC), 2011, pp. 1184-1191

[7]Zhang Pei-ying and Li Cun-he, "Automatic text summarization based on sentences clustering and extraction," in Second IEEE Computer Science and Information technology International Conference, 2009, pp. 167-170

[8] Lesk algorithm - Wikipedia, the free encyclopedia, Available: https://en.wikipedia.org/wiki/Lesk_algorithm