

# Feature Extraction from Comments

Reuel Samuel Sam - 2018103053

V.S.Suryaa - 2018103610

S.Srihari - 2018103601

# Abstract

In this day and age, where the internet has taken control of most people's lives, social media exists as one of the most powerful platforms to share and voice one's opinions. People put up content looking for appreciation and constructive criticism from the general public to help improve their skills in their art. The high volume of user-generated content makes a manual analysis of this discourse unviable. Consequently, automatic analysis techniques are needed to extract the opinions expressed in users' comments, given that these opinions are an implicit parameter of unquestionable interest for a vast section of the society. Our project aims to automate this task of analysing the reactions on the posts and generate a report based on the outcome. Viewers of the post are provided a feature to predict the number of likes their comment would yield based on the post and existing comments.

**Keywords or phrases:** Compiler Design, Python, Comment analysis, Neural Network, Database, Lexical Analysis

# Literature Survey

S.No	Author-Publication-Year	Methodology	Advantages	Limitations
1.	Rajdeep Singh, Roshan Bagla, Harkiran Kaur - 2015	Lexicon based sentiment analysis to analyse the emotion in the comments and provide statistics of likes and dislikes.	Proposes a methodology to identify the polarity of popularity without much hassle and effort. The work also takes into consideration the overall impact of different parts of speeches in grammar.	Takes a very rudimentary approach to analyse the sentiment by taking same polarity for a number of words and thus not indulging the intricacies. Assumes that the user interacts with proper grammar and avoids spelling check.
2.	A.Moreo, M.Romero, J.L.Castro, J.M.Zurita - 2012	Mainly composed of two modules – Focus Detection and Sentiment Analysis. It analyzes each sentence in the comments to create a set of tuples that abstractly represents users' opinions by assigning sentiment expressions to opinion focuses.	Contains colloquial expressions that are of crucial importance to analysis. Hierarchical structure to improve results of focus detection.	Requirement of sufficient information is a must. Also unable to detect hints of sarcasm and satire in comments.

# Literature Survey

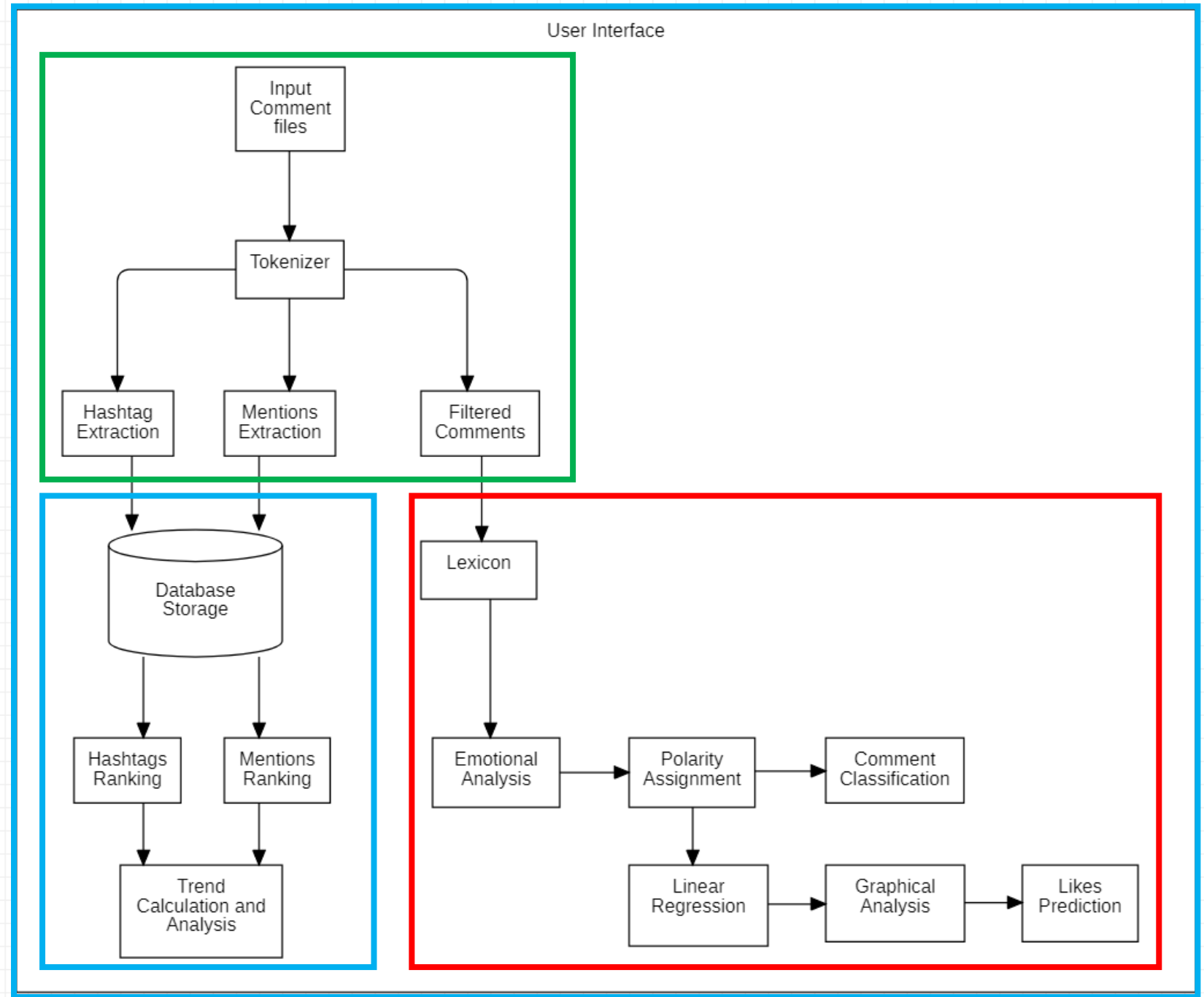
S.No	Author-Publication-Year	Methodology	Advantages	Limitations
3.	Dilesh Tanna, Manasi Dudhane, Amrut Sardar - 2020	Adopted LICW – text analysing software along with SVM, maximum entropy and Naïve Bayes models	Notifies close circle and doctors about the extreme emotions of the user based on his reactions to posts thereby ensuring that improper decisions aren't taken.	The users are individually rated based on their response to other posts. The impact of an individual post and its response hasn't been taken care of.
4.	Anees Ul Hassan, Jamil Hussain, Musarrat Hussain, Muhammad Sadiq, Sungyoung Lee - 2017	Analyzed the different cultural variations, take out emotions, and obtained the sentiment behind ML techniques, comparing them on sentence-level for depression measurement.	Proposed a multi-class emotion classifier that determines emotions with greater precision.	Meta Learning tends to be inaccurate and is problematic with timing and maintenance with complex models
5.	Persia.F, D'Auria.D - 2017	Surveying opportunities and challenges in Online Social Networking with emphasis on the most popular users.	Receiving personalized posts based on liked ones. Focus on semantic security and coping up phishing attacks with celebrities	Combining the views of the close circle general public, makes the media uninviting. Ways to filter responses from comments not discussed.

# Problem Statement

Tackling a domain as wide as social media and its influence comes with a lot of preparation. The Literature Survey gives a description of the issues faced by the previous approaches to this problem. To overcome the issues of need of proper grammar, analysis of comments one by one, detection of sarcasm etc. we have resorted to 'Afinn' that attempts to tackle these issues. Furthermore, to add our own originality to this issue, we have included more features such as like prediction, displaying highest tag count, count occurrence etc.

# Basic Flow Diagram

— Srihari.S  
— Reuel Sam  
— V.S.Suryaa



# Module 1: Tokenizer

- **Input:**

The comments in a post “x” are stored in a file x.txt. The likes obtained by the comment are also entered by the user in the same file.

- **Intermediate Results:**

Module-1 is composed of three phases:

- i. Extraction of Mentions
- ii. Extraction of Hashtags
- iii. Extraction of Filtered Comments

# Module 1

## Phase 1: Extraction of Mentions

- An account is considered mentioned if it is of the form **@account\_name**
- Output:
  - A file x\_mentions.txt is created which contains the list of accounts that have been mentioned in a comment of post “x”.
  - The details in x\_mentions.txt are stored in the format:  
**PostID;account\_name;CommentNumber**

```
letter [a-zA-Z_.]
digit [0-9]
number {digit}+
id {letter}({letter}|{digit})*
delim [ \t]
tag ({letter}|{digit})+
%%
^{number}"LIKES" ;
[ ^{letter}|^{digit}]"@"{tag}{delim} {
    bzero(res,sizeof(res));
    yless(yyleng-1);
    strncpy(res,yytext+2,yyleng-2);
    fprintf(yyout,"%s;%s;%d\n",file_name,res,line_count);
    REJECT;
}
"\n" ;
"\n." ;
"\n".*" \n" {line_count++;REJECT;}
.;
```

Definition of a mention

Keeps track of comment number



# Module 1

## Phase 1: Extraction of Mentions

Output File: 3\_mentions.txt

```
srihari@LAPTOP-AJMTFS87: ~/compiler_design/mini_project/iteration2
srihari@LAPTOP-AJMTFS87:~/compiler_design/mini_project/iteration2$ cat 3.txt
1LIKES The #best day in Karans show!!! @kj
2LIKES @stjobs had better plans with @har@n12 @less1_lie #founders23 APPLE
6LIKES #design #rules the world isn't it @sammer@ #undou#bted come on agree Sameer?
24LIKES @reuel @suryaa fun fact it is #necessary to have a #space #endl after each comment....
0LIKES Nlp well that was on #purpose start of sentence!!!
.
srihari@LAPTOP-AJMTFS87:~/compiler_design/mini_project/iteration2$ lex mentions.l
srihari@LAPTOP-AJMTFS87:~/compiler_design/mini_project/iteration2$ g++ lex.yy.c -ll
srihari@LAPTOP-AJMTFS87:~/compiler_design/mini_project/iteration2$ ./a.out 3.txt
srihari@LAPTOP-AJMTFS87:~/compiler_design/mini_project/iteration2$ cat 3_mentions.txt
3;kj;1
3;stjobs;2
3;less1_lie;2
3;reuel;4
3;suryaa;4
```

# Module 1

## Phase 2: Extraction of Hashtags

- Any hashtag is of the form **#word**
- Output:
  - A file x\_hashes.txt is created which contains the list of accounts that have been mentioned in a comment of post “x”.
  - The details in x\_hashes.txt are stored in the format:

**PostID;word;CommentNumber**

```
letter [a-zA-Z_]
digit [0-9]
number {digit}+
id {letter}({letter}|{digit})*
tag ({letter}|{digit})+
delim [ \t]

%%
^{number}"LIKES" ;

[ \n]"#{tag}{delim} {
    bzero(res,sizeof(res));
    yylless(yyleng-1);
    strncpy(res,yytext+2,yyleng-2);
    fprintf(yyout,"%s;%s;%d\n",file_name,res,line_count);
    REJECT;
}
"\n" ;
"\n." ;
"\n".*"\n" {line_count++;REJECT;}
. ;
%%
```

Definition of a hashtag

# Module 1

## Phase 2: Extraction of Hashtags

Output File: 3\_hashes.txt

```
srihari@LAPTOP-AJMTFS87: ~/compiler_design/mini_project/iteration2
srihari@LAPTOP-AJMTFS87:~/compiler_design/mini_project/iteration2$ cat 3.txt
1LIKES The #best day in Karans show!!! @kj
2LIKES @stjobs had better plans with @har@n12 @less1_lie #founders23 APPLE
6LIKES #design #rules the world isn't it @sammer@ #undou#bted come on agree Sameer?
24LIKES @reuel @suryaa fun fact it is #necessary to have a #space #endl after each comment....
0LIKES Nlp well that was on #purpose start of sentence!!!
```

```
srihari@LAPTOP-AJMTFS87:~/compiler_design/mini_project/iteration2$ lex hashes.l
srihari@LAPTOP-AJMTFS87:~/compiler_design/mini_project/iteration2$ g++ lex.yy.c -ll
srihari@LAPTOP-AJMTFS87:~/compiler_design/mini_project/iteration2$ ./a.out 3.txt
srihari@LAPTOP-AJMTFS87:~/compiler_design/mini_project/iteration2$ cat 3_hashtags.txt
3;best;1
3;founders23;2
3;design;3
3;rules;3
3;necessary;4
3;space;4
3;endl;4
3;purpose;5
```


# Module 1

## Phase 3: Extraction of Filtered Comments

- Output:

- A file `x_filtered_comments.txt` is created which contains the comments of post “x” filtered by removing mentions, punctuations, proper nouns and stores the likes obtained by that comment.
- The details in `x_filtered_comments.txt` are stored in the format:  
**LikesObtained;filteredComment;CommentNumber**

Storing the count of likes obtained by a comment



```
^{number}"LIKES "{
    int i=0;
    while(yytext[i]!='L'){
        fprintf(yyout,"%d",yytext[i++]-48);
    }
    fprintf(yyout,";");
    yylless(yyleng-1);
}
```

# Module 1

## Phase 3: Extraction of Filtered Comments

### Elimination of proper nouns

- Criteria for a proper Noun:

**If((word[0] >= 'A' && word[0] <= 'Z' && doesn't start a sentence) || (word.toUpperCase() == word))  
return true;**

```
letter [a-zA-Z_.]
digit [0-9]
number {digit}+
id {letter}({letter}|{digit})*
tag ({letter}|{digit})+
delim [ \t]
hash [ \n]"#{tag}{delim}
mention [^{letter}|^{digit}]"@{tag}{delim}
punctuation [.!?'`,:#@]+
caps [ ][A-Z]+[ ]
first [ ][A-Z][a-z]+{punctuation}[ ]
```

```
{mention} {
    yyless(yyval-1);
}
{caps} {
    fprintf(yyout, " ");
    yyless(yyval-1);
}
{first} {
    yyless(yyval-1);
}
{punctuation} {
    fprintf(yyout, " ");
}
```

# Module 1

## Phase 3: Extraction of Filtered Comments

Output File: 3\_hashes.txt

```
srihari@LAPTOP-AJMTFS87: ~/compiler_design/mini_project/iteration2
srihari@LAPTOP-AJMTFS87:~/compiler_design/mini_project/iteration2$ cat 3.txt
1LIKES The #best day in Karans show!!! @kj
2LIKES @stjobs had better plans with @har@n12 @less1_lie #founders23 APPLE
6LIKES #design #rules the world isn't it @sammer@ #undou#bted come on agree Sameer?
24LIKES @reuel @suryaa fun fact it is #necessary to have a #space #endl after each comment....
0LIKES Nlp well that was on #purpose start of sentence!!!

srihari@LAPTOP-AJMTFS87:~/compiler_design/mini_project/iteration2$ lex filtered_comments.l
srihari@LAPTOP-AJMTFS87:~/compiler_design/mini_project/iteration2$ g++ lex.yy.c -ll
srihari@LAPTOP-AJMTFS87:~/compiler_design/mini_project/iteration2$ ./a.out 3.txt
srihari@LAPTOP-AJMTFS87:~/compiler_design/mini_project/iteration2$ cat 3_filteredComments.txt
1; The  best day in Karans show  ;1
2; had better plans with  har n12  founders23  ;2
6;  design  rules the world isn't it  sammer  undou bted come on agree ;3
24; fun fact it is  necessary to have a  space  endl after each comment  ;4
0; Nlp well that was on  purpose start of sentence  ;5
srihari@LAPTOP-AJMTFS87:~/compiler_design/mini_project/iteration2$
```



# Module 1: Intermediate Results

- i. **Input File:** 3.txt (file containing the comments in user's post along with the like count for each comment)
- ii. **Output Files:**
  - 3\_mentions.txt
  - 3\_hashtags.txt
  - 3\_filteredComments.txt
- iii. These output files are passed on to the Module - 2

```
srihari@LAPTOP-AJMTFS87: ~/compiler_design/mini_project/iteration2
srihari@LAPTOP-AJMTFS87:~/compiler_design/mini_project/iteration2$ cat 3.txt
1LIKES The #best day in Karans show!!! @kj
2LIKES @stjobs had better plans with @har@n12 @less1_lie #founders23 APPLE
6LIKES #design #rules the world isn't it @sammer@ #undou#bted come on agree Sameer?
24LIKES @reuel @suryaa fun fact it is #necessary to have a #space #endl after each comment....
0LIKES Nlp well that was on #purpose start of sentence!!!

srihari@LAPTOP-AJMTFS87:~/compiler_design/mini_project/iteration2$ lex mentions.l
srihari@LAPTOP-AJMTFS87:~/compiler_design/mini_project/iteration2$ g++ lex.yy.c -ll
srihari@LAPTOP-AJMTFS87:~/compiler_design/mini_project/iteration2$ ./a.out 3.txt
srihari@LAPTOP-AJMTFS87:~/compiler_design/mini_project/iteration2$ cat 3_mentions.txt
3;kj;1
3;stjobs;2
3;less1_lie;2
3;reuel;4
3;suryaa;4
srihari@LAPTOP-AJMTFS87:~/compiler_design/mini_project/iteration2$ lex hashes.l
srihari@LAPTOP-AJMTFS87:~/compiler_design/mini_project/iteration2$ g++ lex.yy.c -ll
srihari@LAPTOP-AJMTFS87:~/compiler_design/mini_project/iteration2$ ./a.out 3.txt
srihari@LAPTOP-AJMTFS87:~/compiler_design/mini_project/iteration2$ cat 3_hashtags.txt
3;best;1
3;founders23;2
3;design;3
3;rules;3
3;necessary;4
3;space;4
3;endl;4
3;purpose;5
srihari@LAPTOP-AJMTFS87:~/compiler_design/mini_project/iteration2$ lex filtered_comments.l
srihari@LAPTOP-AJMTFS87:~/compiler_design/mini_project/iteration2$ g++ lex.yy.c -ll
srihari@LAPTOP-AJMTFS87:~/compiler_design/mini_project/iteration2$ ./a.out 3.txt
srihari@LAPTOP-AJMTFS87:~/compiler_design/mini_project/iteration2$ cat 3_filteredComments.txt
1; The best day in Karans show ;1
2; had better plans with har n12 founders23 ;2
6; design rules the world isn't it sammer undou bted come on agree ;3
24; fun fact it is necessary to have a space endl after each comment ;4
0; Nlp well that was on purpose start of sentence ;5
srihari@LAPTOP-AJMTFS87:~/compiler_design/mini_project/iteration2$
```

# Module 2: Database Storage

- **Input**





- Three files from output of Module 1
  - #\_mentions.txt
  - #\_hashtags.txt
  - #\_filteredcomments.txt

- **Output**





- Stores the file contents in database in appropriate tables
- Calls Module 3 along with #\_filteredcomments.txt to run the analysis



# Module 2: Populating Database

Table: hashtags     New Record Delete Record

content	commentno	postno
Filter	Filter	Filter

Table: mentions     New Record Delete Record

content	commentno	postno
Filter	Filter	Filter

Initially both hashtags and mentions tables are empty as shown above.

# Module 2: Populating Database (Contd.)

- Example File: 3\_mentions.txt and 3\_hashtags.txt
- Code for updating and populating:

```
def updatedatabase(conn, List, table):
    c = conn.cursor()
    for i in range(len(List)):
        cursor = conn.execute("Select * from %s where content = '%s'"%(table, List[i].content))
        flag = False
        for row in cursor:
            if row[0].lower() == List[i].content.lower() and row[1] == List[i].cno and row[2] == List[i].cn:
                flag = True
                break
        else:
            flag = False

        if flag == True:
            pass
            #print("Duplicate comment: ",row)
        else:
            c.execute("INSERT INTO %s VALUES('%s',%d,%d)"%(table,List[i].content.lower(),int(List[i].cno),int(List[i].cn)))
            #print("Record inserted")

    conn.commit()
```

# Module 2: Populating Database (Contd.)

- Example File: 3\_mentions.txt and 3\_hashtags.txt
- Database after population:

Table: hashtags ↕ ↺ ↻ 🖨 📄 New Record Delete Record

	content	commentno	postno
	Filter	Filter	Filter
1	best	1	3
2	founders23	2	3
3	design	3	3
4	rules	3	3
5	necessary	4	3
6	space	4	3
7	endl	4	3
8	purpose	5	3

Table: mentions ↕ ↺ ↻ 🖨 📄 New Record Delete Record

	content	commentno	postno
	Filter	Filter	Filter
1	stjobs	2	3
2	less1_lie	2	3
3	reuel	4	3
4	suryaa	4	3
5	suryaa	5	3

# Module 2: Finding count

- Code:

```
def countoccurance(conn, key, table):  
    c = conn.cursor()  
    res = c.execute("SELECT COUNT(content) FROM %s WHERE content='%s'"%(table, key.lower()))  
    for row in res:  
        return row[0]  
    else:  
        return 0
```

*Note: Output shown and described in Module 4*

# Module 2: Finding highest count

- Code:

```
def findmax(conn,table):
    c = conn.cursor()
    res = c.execute("SELECT DISTINCT content, COUNT(content) as count FROM %s GROUP BY content ORDER BY
count_list = list()
    flag = False
    for row in res:
        flag = True
        count_list.append(row)
    if flag == False:
        count_list.append(None)
    return -1

table = tabulate(count_list, headers=["Title","Count"])
print(table)
return count_list[0], table
```

*Note: Output shown and described in Module 4*

# Module 3: Overview

## **Input:**

A text file containing the filtered comment with number of likes and serial number of comments

## **Phase-1 goal:**

Emotional analysis of the filtered comments using lexicon to extract crux emotion of the post

## **Phase-2 goal:**

Predicting the number of likes for a comment with features observed and extracted in previous phase

# Module 3: What is afinn lexicon?

- AFINN sentiment analysis in Python: Wordlist-based approach for sentiment analysis.
- Developed and curated by Finn Nielsen, you can find more details on this lexicon in the paper, *“A new ANEW: evaluation of a word list for sentiment analysis in microblogs”*, *proceedings of the ESWC 2011 Workshop*.
- The current version of the lexicon is ***AFINN-en-165.txt*** and it contains over 3,300+ words with a polarity score associated with each word.

# Module 3: Generating score for comments

comments.txt :

```
# no of likes; filtered comment; serial number
1; The best day in Karans show ;1
2; had better plans with har n12 founders23 ;2
```

```
from afinn import Afinn
import csv

# text file containing filtered comments
file1 = open('comments.txt', 'r')
Lines = file1.readlines()

# neural network used for emotion analysis
analyze = Afinn()

row = []

for line in Lines:
    # extracting data required for analysis
    output = line.split(';')
    # polarity of each word is analyzed to find the overall emotion of comment
    score = analyze.score(output[1])
```



# Module 3: Intermediate result for Phase 1

```
# generating intermediate result
print("Comment:" + output[1].strip() + "\nScore:" + str(score))
```

Result:

```
Comment:The best day in Karans show
Score:3.0
Comment:had better plans with har n12 founders23
Score:2.0
Comment:design rules the world isn't it sammer undou bted come on agree
Score:1.0
Comment:fun fact it is necessary to have a space endl after each comment
Score:4.0
Comment:Nlp well that was on purpose start of sentence
Score:-2.0
Comment:great show incredible loving it
Score:5.0
Comment:yay
Score:0.0
Comment:features are important take actions immediately
Score:2.0
Comment:excellent display of talent wow
Score:9.0

Process finished with exit code 0
```

# Module 3: Preparing data for Phase-2

We compiled the previously obtained data and included the score predicted using neural network.

```
# for each comment
l1 = [output[0], output[1], output[2].strip(), score]
# compiling data for all comment
row.append(l1)
```

We write the data onto an output file ('output.txt')

```
# output file for phase 1
filename = "output.csv"
# categorizing and writing data
fields = ['no-of-likes', 'comments', 'serial', 'score']
# writing to csv file
with open(filename, 'w') as csvfile:
    # creating a csv writer object
    csvwriter = csv.writer(csvfile)

    # writing the fields
    csvwriter.writerow(fields)

    # writing the data rows
    csvwriter.writerows(row)
```

# Module 3: Pre-processing data for Phase 2

Output.csv:

```
no-of-likes, comments, serial, score
1, The best day in Karans show ,1, 3.0
32, great show incredible loving it ,6, 5.0
```

```
import numpy as np
import matplotlib.pyplot as plt # To visualize
import pandas as pd # To read data
from sklearn.linear_model import LinearRegression

# load data set
data = pd.read_csv('output.csv')

# features to fit the linear regression
X = data.iloc[1:, 0].values.reshape(-1, 1) # values converts it into a numpy array
Y = data.iloc[1:, 3].values.reshape(-1, 1) # -1 means that calculate the dimension of rows, but have 1 column
```

# Module 3: Intro to Linear Regression

- In statistics, linear regression is a linear approach to modeling the relationship between a scalar response (or dependent variable) and one or more explanatory variables (or independent variables).
- Linear regression is a linear model, e.g. a model that assumes a linear relationship between the input variables ( $x$ ) and the single output variable ( $y$ ). More specifically, that  $y$  can be calculated from a linear combination of the input variables ( $x$ ).
- We choose X-axis to be the score for each comment predicted using the neural network and Y-axis to be the number of likes obtained in the comment .

# Module 3: Training and visualizing

- We use `LinearRegression()` class available in sklearn library to train the data
- We then predict using the training data
- The plot of predicted and real data gives the insight into the comment trend

```
linear_regressor = LinearRegression() # create object for the class Linear Regression loaded using sklearn

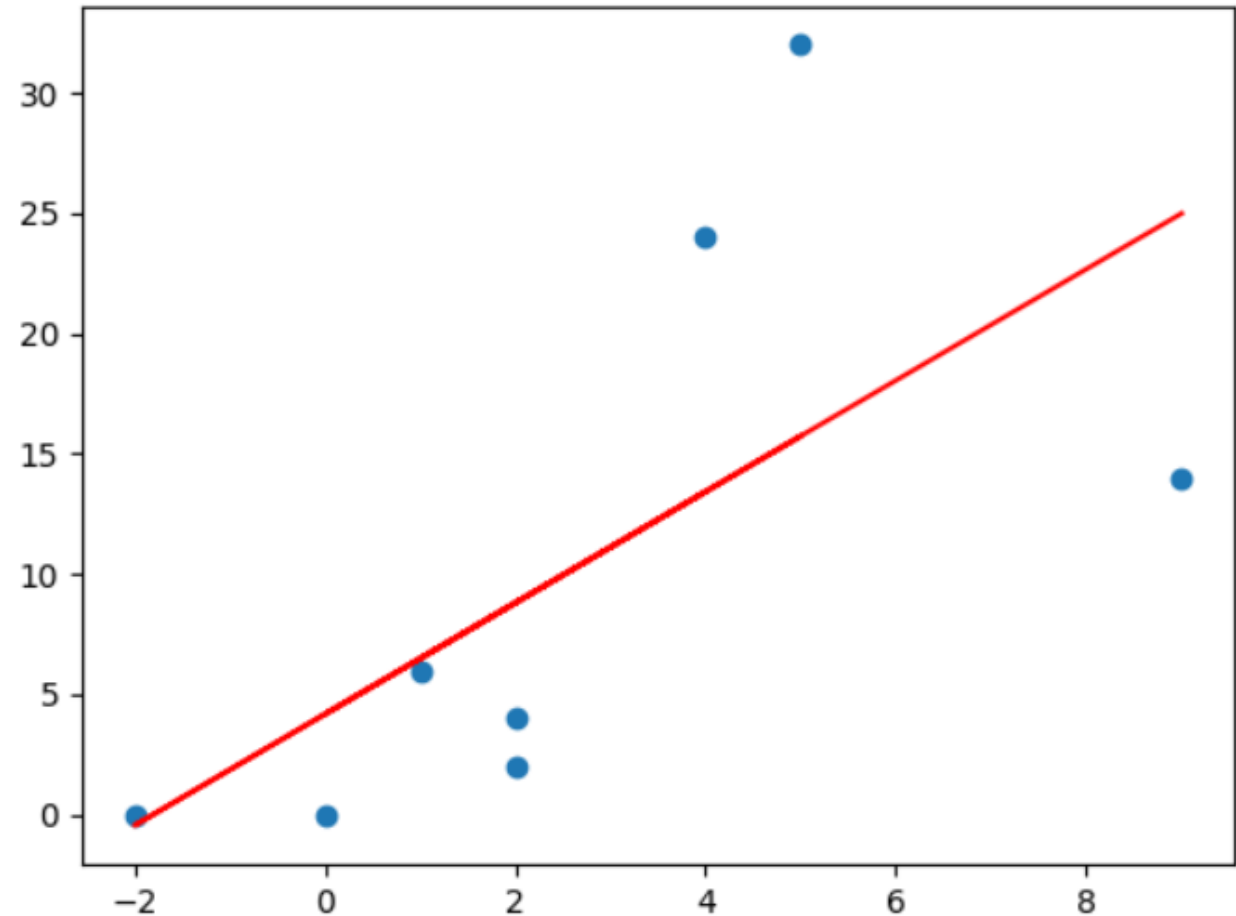
# training using linear regression
linear_regressor.fit(X, Y) # perform linear regression
Y_pred = linear_regressor.predict(X) # make predictions

# visualization of training data and prediction
plt.scatter(X, Y)
plt.plot(X, Y_pred, color='red')
plt.show()
```

# Module 3: Visualizing intermediate result

Score	Actual Likes	Predicted Likes
2	2	8.80438
1	6	6.49139
4	24	13.4304
-2	0	-0.447574
5	32	15.7433
0	0	4.1784
2	4	8.80438
9	14	24.9953

X-axis: Score Y-axis: No of likes



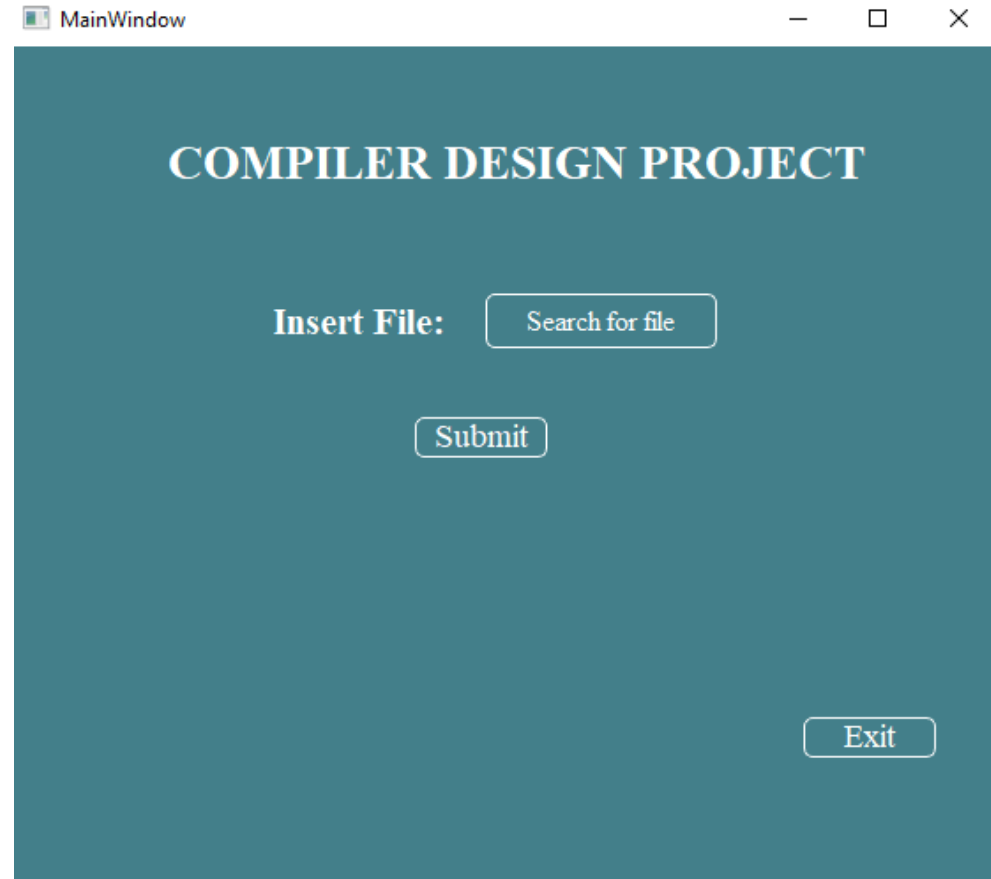
# Module 4: Interface

- This module focuses on combining all the modules and presenting it at a level of abstraction and ease of use to the user
- Built on Python, it makes use of PyQt5 package for the interface
- Each module is called by the interface depending on user's wish.

# Module 4: Introductory Page

*Uses Module 1*

- File entered by user is of the format specified in module 1
- On submission, module 1 is called and its responsibilities are carried out.

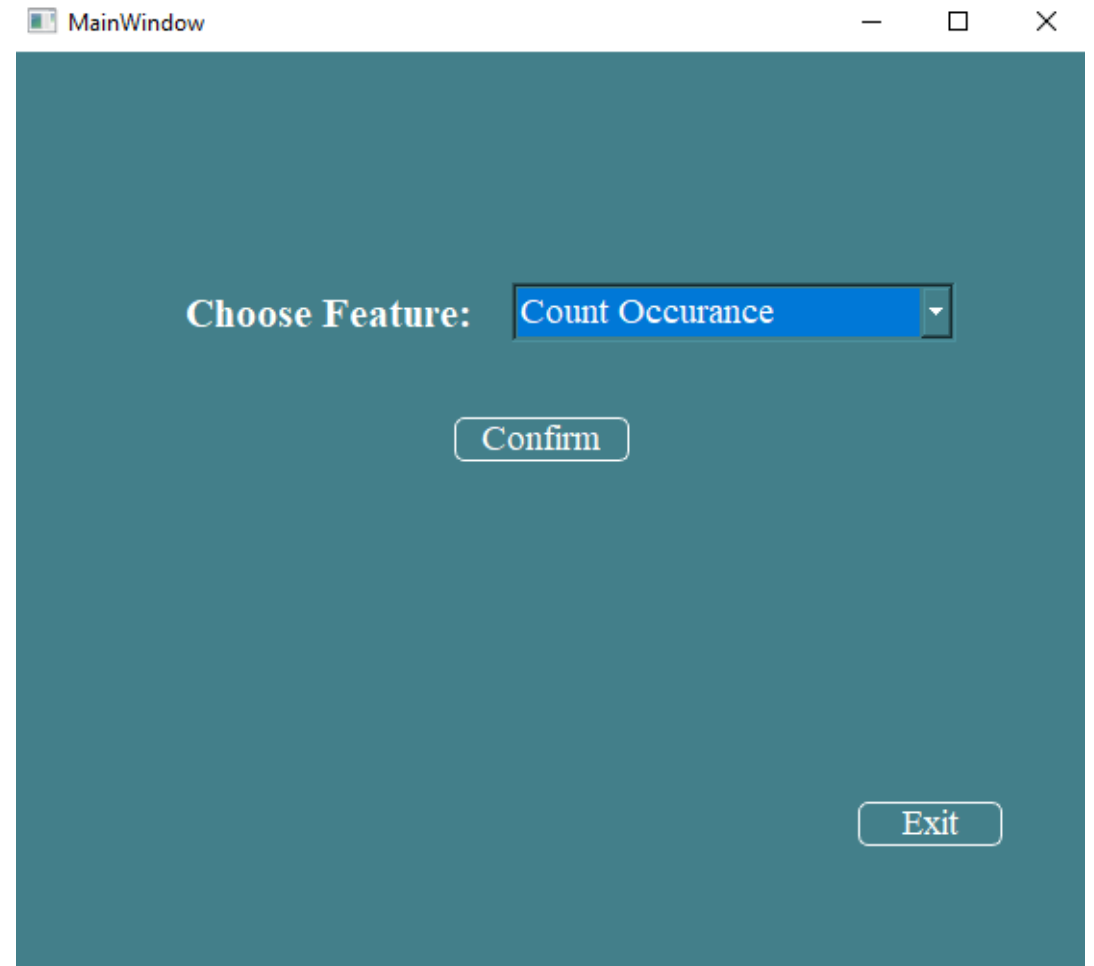
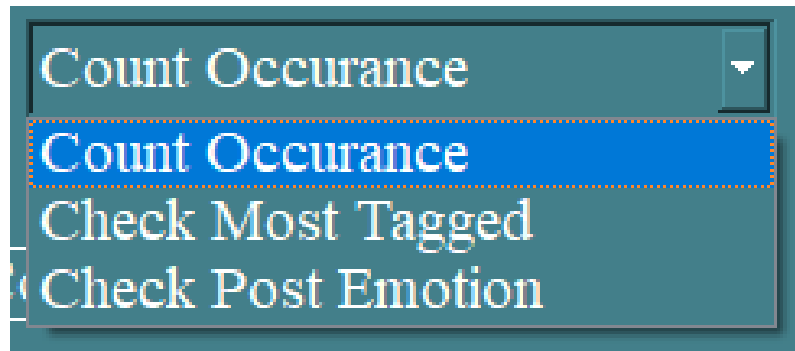




# Module 4: Features Page

*Uses Module 2*

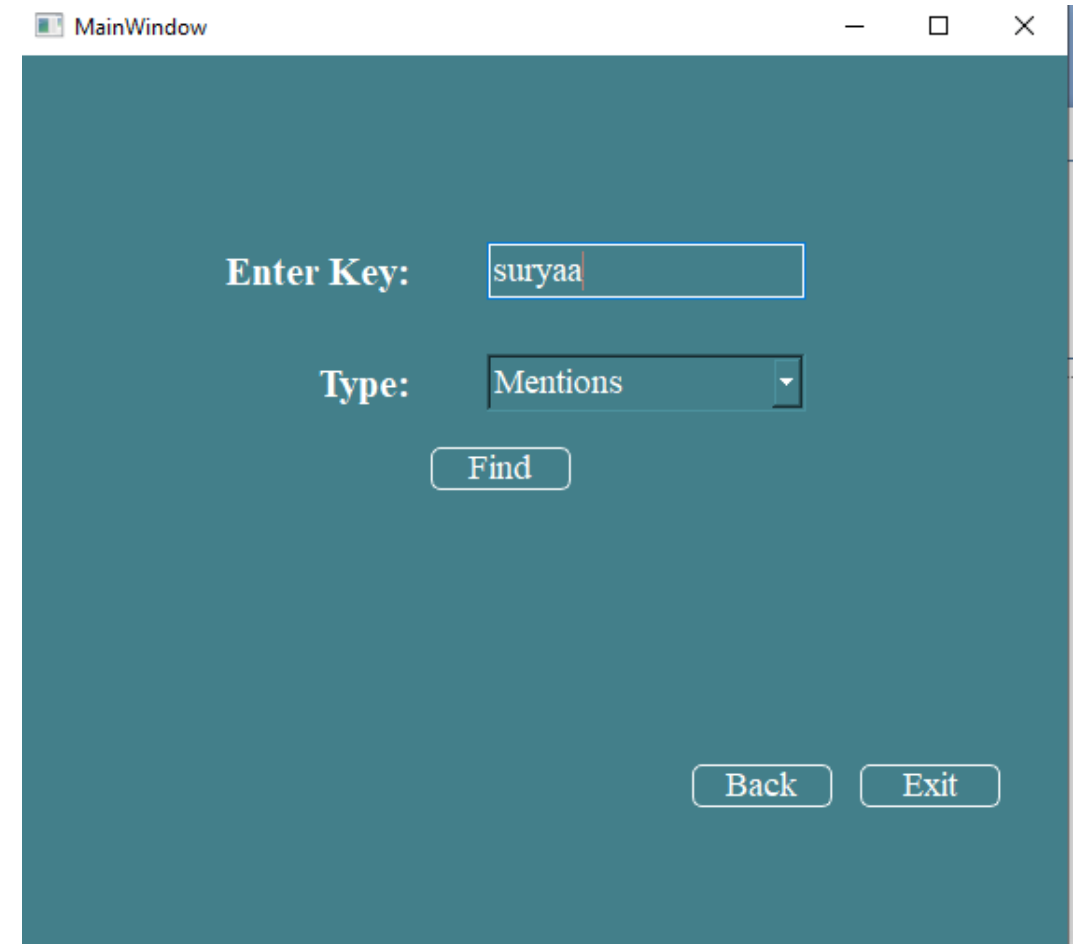
- This allows user to make use of the features incorporated in our project
- The feature options are:



# Module 4: Count Occurrence

*Uses Module 2*

- Key string is then searched for within the database
- Type specified informs the interface whether the user is looking for a mention or a hashtag.



MainWindow

Enter Key:

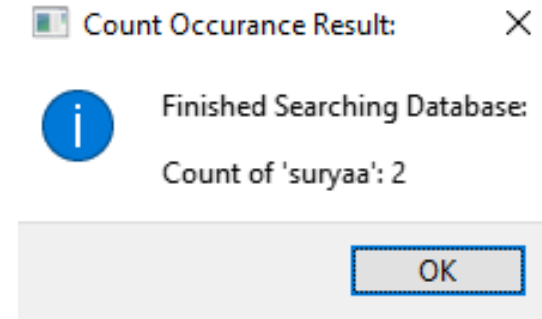
Type:

Find

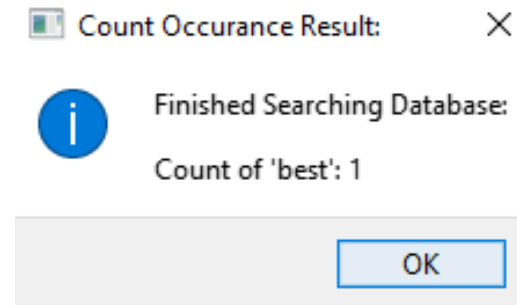
Back Exit

# Module 4: Count Occurrence (Contd.) *Uses Module 2*

- Mention Search: key 'suryaa'



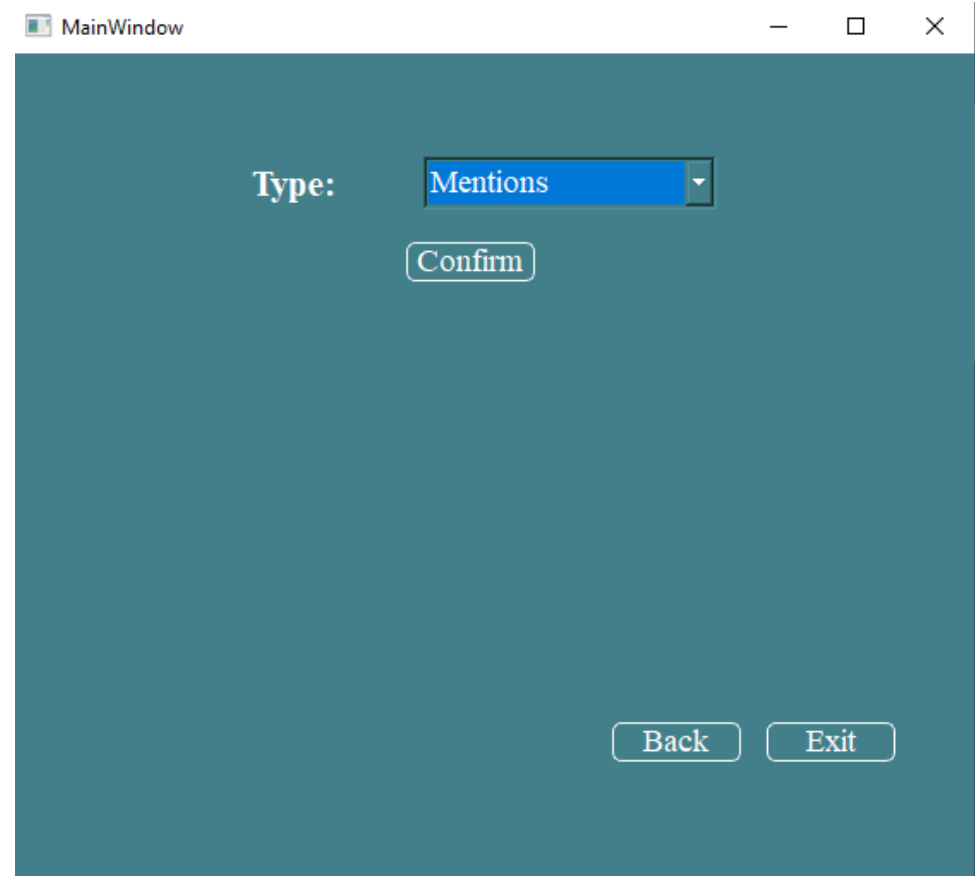
- Hashtag Search: key 'best'



# Module 4: Most tagged

- User can check which mention or hashtag has been tagged the most on their post.
- Makes use of functions specified in module 2

*Uses Module 2*



# Module 4: Most tagged (Contd.)

*Uses Module 2*

- Most tagged Mention

Most Tagged Result: ×

**i** Finished Searching Database:  
Maximum Occuring mentions is 'suryaa' with count of 2

OK Hide Details...

Title	Count
suryaa	2
stjobs	1
reuel	1
less1_lie	1

- Most tagged Hashtag

Most Tagged Result: ×

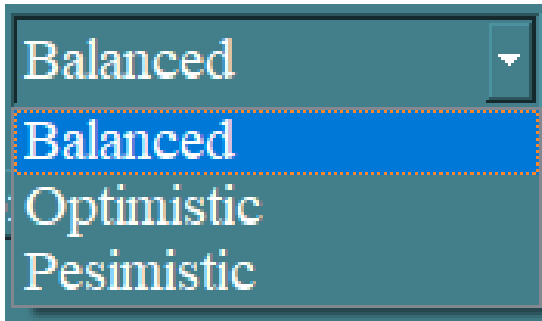
**i** Finished Searching Database:  
Maximum Occuring hashtags is 'space' with count of 1

OK Hide Details...

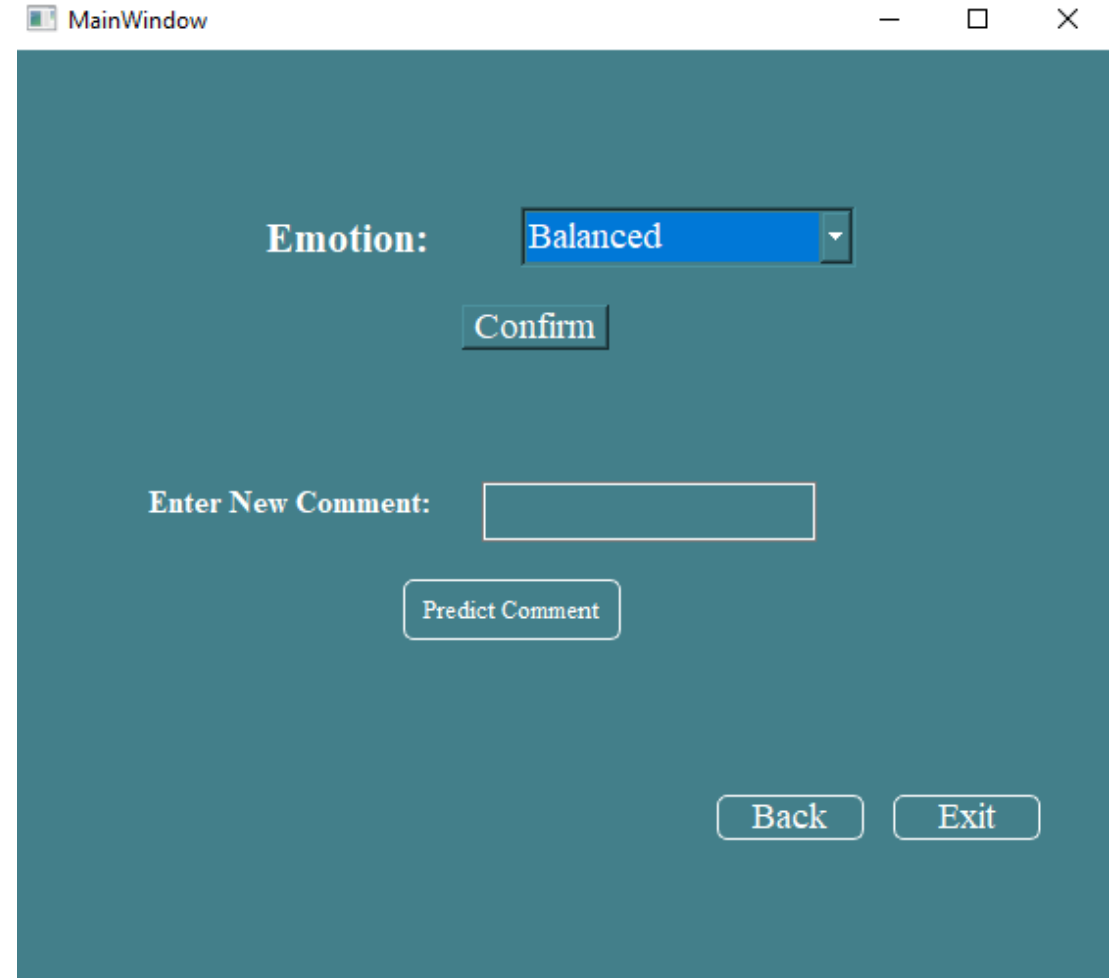
Title	Count
space	1
rules	1
purpose	1
necessary	1

# Module 4: Post Emotion

- User can display the different comments based on emotion portrayed.
- Emotions identified:

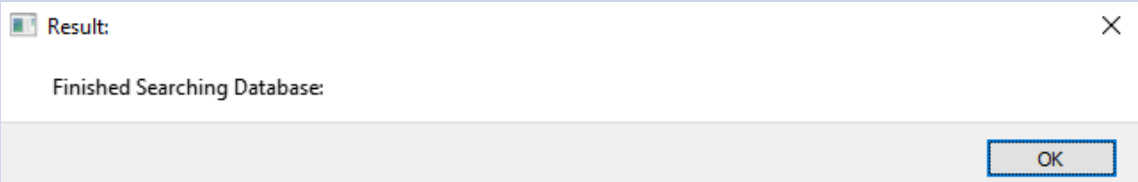
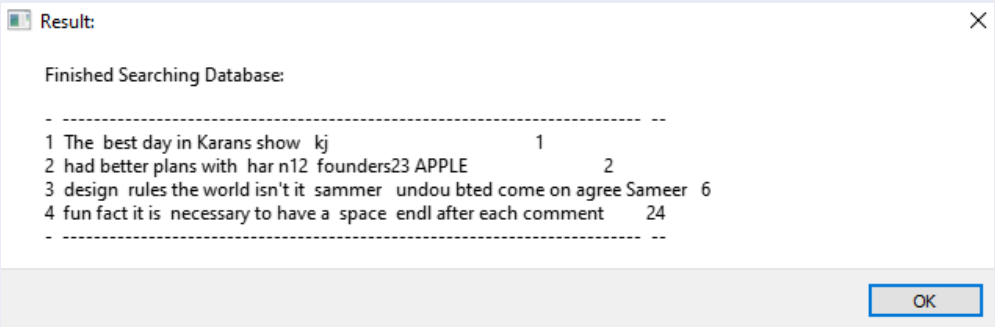
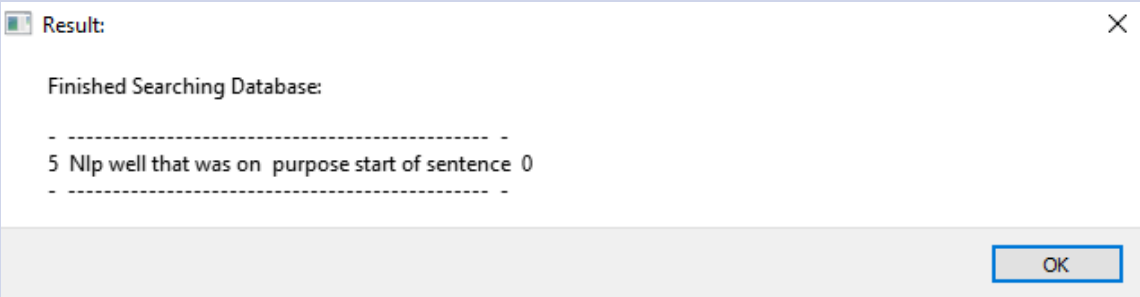


*Uses Module 3*



# Module 4: Post Emotion (Contd.)

Uses Module 3

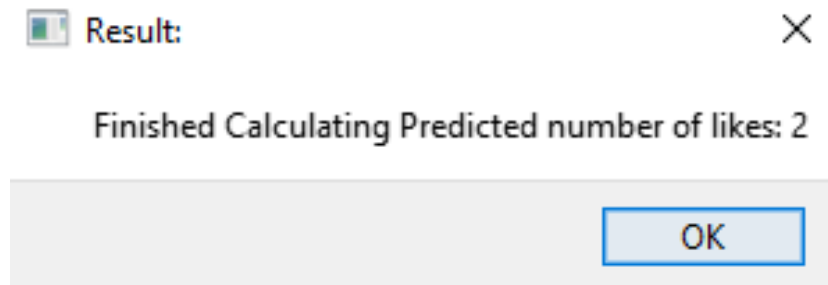
Emotion	Output
Balanced	
Optimistic	
Pesimistic	

# Module 4: Post Emotion (Contd.)

*Uses Module 3*

- Here, a new comment is entered and the number of likes it will fetch is predicted

Enter New Comment:





# References

- K. M. Anitha, V. Saraswathy, K. Gayathri and S. S. Priya, "An Approach To Comment Analysis In Online Social Media," 2019 3rd International Conference on Computing and Communications Technologies (ICCCT), Chennai, India, 2019, pp. 332-335, doi: 10.1109/ICCCT2.2019.8824949.
- A. U. Hassan, J. Hussain, M. Hussain, M. Sadiq and S. Lee, "Sentiment analysis of social networking sites (SNS) data using machine learning approach for the measurement of depression," 2017 International Conference on Information and Communication Technology Convergence (ICTC), Jeju, 2017, pp. 138-140, doi: 10.1109/ICTC.2017.8190959.
- Persia, F., & D'Auria, D. (2017). A Survey of Online Social Networks: Challenges and Opportunities. 2017 IEEE International Conference on Information Reuse and Integration (IRI).
- arXiv:1103.2903 - A new ANEW: Evaluation of a word list for sentiment analysis in microblogs - Finn Årup Nielsen
- A. U. Hassan, J. Hussain, M. Hussain, M. Sadiq and S. Lee, "Sentiment analysis of social networking sites (SNS) data using machine learning approach for the measurement of depression," 2017 International Conference on Information and Communication Technology Convergence (ICTC), Jeju, 2017, pp. 138-140, doi: 10.1109/ICTC.2017.8190959.
- X. Lin, T. Bu, Q. He and Y. Liang, "A study on the impact of positive and negative comments on consumers' perceived usefulness," 2019 16th International Conference on Service Systems and Service Management (ICSSSM), Shenzhen, China, 2019, pp. 1-6, doi: 10.1109/ICSSSM.2019.8887617.
- R. Singh, R. Bagla and H. Kaur, "Text analytics of web posts' comments using sentiment analysis," 2015 International Conference and Workshop on Computing and Communication (IEMCON), Vancouver, BC, 2015, pp. 1-5, doi: 10.1109/IEMCON.2015.7344534.

# FUTURE SCOPE OF THE PROJECT

- We are planning on analysing further to find dislikes to remove ambiguity of likes (Users who make a negative comment and like the post)
- Taking a deeper approach to the project by finding adjective , adverb and other part of speeches and normalizing the polarity of the word accordingly. This is taken into consideration because some part of speeches tend to make a stronger impact on meaning than others.
- Immediate plan on further developing the User Interface design
- Longer vision ahead is to build a small scale social media platform and let users interact and extract further details to make an in-depth analysis.