**SENTIMENTAL ANALYSIS OF TEXT DOCUMENTS USING LEXICAL ANALYSIS**

PRACTICAL ASSIGNMENT REPORT

SUBMITTED TO

**PARKAVI A.**

RAMAIAH INSTITUTE OF TECHNOLOGY

(Autonomous Institute, Affiliated to VTU)

Bangalore – 560054

**SUBMITTED BY**

**Atif Adib(1ms14cs023)**

**Anusha Sairam(1ms14cs022)**

**Aditya Barsainya(1ms14cs008)**

As part of the Course **Compiler Design – CS1561/CS612**

SUPERVISED BY

Faculty Name

**PARKAVI A.**



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

RAMAIAH INSTITUTE OF TECHNOLOGY

Jan-May 2017

Department of Computer Science and Engineering

Ramaiah Institute of Technology

(Autonomous Institute, Affiliated to VTU)

Bangalore – 54



**CERTIFICATE**

This is to certify that **Atif Adib(1ms14cs023),Anusha Sairam(1ms14cs022),Aditya Barsainya(1ms14cs008)** have completed the **“SENTIMENTAL ANALYSIS OF A TEXT DOCUMENT USING LEXICAL ANALYSIS”** as part of practical assignment.

We declare that the entire content embodied in this B.E. 6th Semester report contents are not copied.

Submitted by **Guided by**

**1. ATIF ADIB**

**2. ANUSHA SAIRAM**

**3. ADITYA BARSAINYA**

**Prof. Parkavi A.**

(Dept of CSE, RIT) (Assistant Professor, Dept. of CSE, RIT)

Department of Computer Science and Engineering

Ramaiah Institute of Technology

(Autonomous Institute, Affiliated to VTU)

Bangalore – 54



**Evaluation Sheet**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Sl. No** | **USN**  **Name** | **Content**  **and Demonstration**  **(5)** | **Speaking Skills**  **(1)** | **Teamwork**  **(1)** | **Neatness and care**  **(1)** | **Effectiveness**  **& Productivity (2)** | **Total Marks**  **(10)** |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

Evaluated By

Name: Parkavi A.

Designation: Assistant Professor

Department: Computer Science & Engineering, RIT

Signature:

**Contents:**

1. Abstract

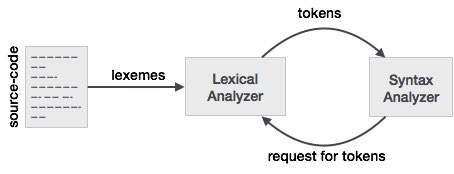
2. Source Code

3. Result Snapshots

4. References

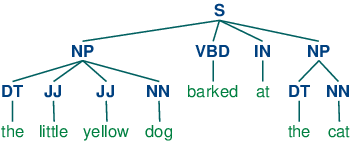
**Abstract:**

Sentimental Analysis of a Text Document using a lexical Analyzer, the goal is to use a lexical analyzer to break the text document down into small chunks and use a corpus to tag each token with its P.O.S (part of speech) tag.



Once the tokens are generated , the grammar of the language in which the book is written is used to generate the parse tree for each sentence.

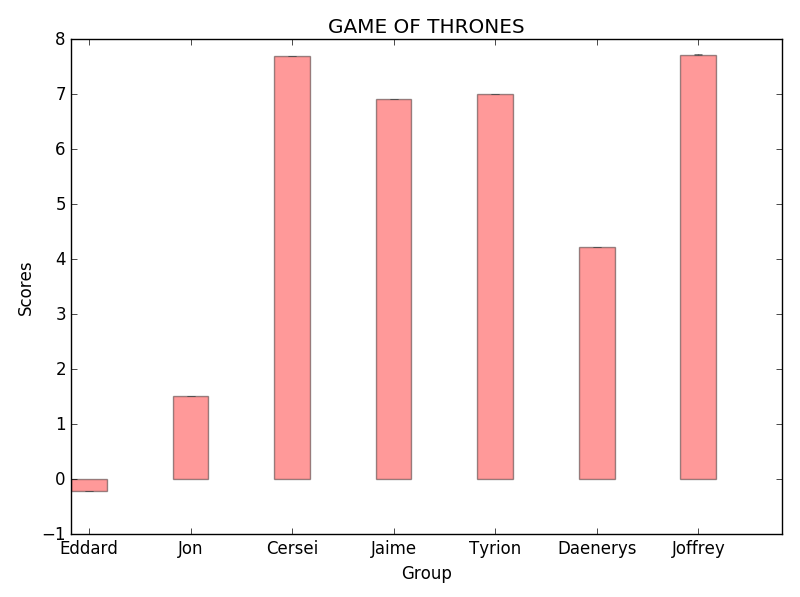
Leaf nodes are then tagged as a word from the sentence and a P.O.S tag.



**Output:**

The Characters from the novel are identified and an adjective list for each character is generated using regular expressions.

Each Character is given a sentimental score on a continuous scale of -1 to 1.



\*\*Higher score indicate a negative character (Antagonist).

\*\*Lower score indicates a positive character

(Protagonist).

**Source Code:**

**P.O.S tagging to Identify Characters**

from textblob import TextBlob

file=open("gameofthrones1.txt","r")

lines=file.readlines()

characters=[]

for l in lines:

obj=TextBlob(l)

for i in obj.tags:

if('NNP' in obj.tags):

characters.append(obj.tags[0])

print(characters)

**Output:**

[‘Arya’,’Cersei’,’Eddard’,’Jaime’,’Joffery’,’Tyrion’]

**Adjective Extraction:**

from textblob import TextBlob

file=open("gameofthrones1.txt","r")

lines=file.readlines()

adjectives=[]

for l in lines:

obj=TextBlob(l)

for i in obj.tags:

if('JJ' in i):

adjectives.append(i[0])

print(adjectives)

OUTPUT:

['complete', 'original', 'boxed', 'electronic', 'mechanical', 'boxed', '978-0-553-89785-2', 'red', 'purple', 'long', 'twelve', 'wyvern', 'ancient', 'uneasy', 'old', 'old', 'comet', 'bright', 'terrible', 'Such', 'black', 'rough', 'old', 'grown', 'lifetime’s', 'hard-won', 'great', 'ignorant', 'pale', 'grey', 'hot', 'white', 'long-expected', 'fearful', 'summer’s', 'many', 'solemn', '“The', 'white', 'great', '“Her', 'old', 'eightieth', 'frail', 'unsteady', 'Last', 'mere', '“Go', 'ill', 'feeble', 'capable', 'thin', 'sure', 'old', 'different', 'clang-a-dang', 'bong-dong', 'blue', 'pretty', 'square', 'unfortunate', 'own', 'half', 'stiff', 'dead', 'black', 'grey', 'white', 'answered', 'next', 'old', 'pleasure.”', 'polite', 'grim', 'lonely', 'wet', 'sent', 'past', 'late', 'restless', 'red', 'patched', 'piebald', 'steep', 'oh.”', 'sorry', 'soft', 'obese', 'subject', 'only', 'only', 'little', 'sad', 'early', 'past', 'bad', '“The', 'olden', 'great', 'fearsome', 'thousand', 'simple', 'small', 'pink', 'own', 'gentle', 'red', 'red', 'daughter’s', 'stern', 'such', '“The', 'sweet', 'brave', 'little', 'white', 'white', 'different', 'heavy', 'other', 'important', 'great', 'last', 'sixteen', 'true', '“In', 'good', 'warm', 'bountiful', 'long', 'such', '“The', 'oh.”', 'dry', 'oh.”', 'impressive', 'white', 'bright', 'black', 'mere', 'truebred', 'white', 'raven', 'table', '“Lady', 'open', '“It', 'few', 'clever', 'clever', '“The', 'other', 'white', '“He', 'scared', 'old', 'cherished', 'narrow', 'king—the', 'old', 'mad', 'days—had', 'splendid', 'fruitless', 'nimble', 'boy', 'lord’s', 'two-masted', 'hundred', 'lady', 'fresh', 'swollen', 'third', 'dead', 'skin', 'white', 'wrinkled', 'wet', 'burial', 'clammy', 'capable', 'fool’s', 'red', 'green', '“The', 'mad', 'old', 'poppy', 'painless', 'many', 'ring-a-ling', 'bong', 'white', '“A', 'anxious', 'last', 'last', 'many', 'few', 'restless', 'old', 'friendly', 'turnpike', 'central', 'guardian', 'black', 'old', 'bad', 'grateful', 'tall', 'arched', 'past', 'triple-decked', 'small', 'big-bellied', '“It’s', 'long', '“You', 'slight', 'low', 'common', 'well-worn', 'green', 'thin', 'brown', 'brown', 'worn', 'small', 'black', 'favorite', 'notorious', 'elusive', 'strong', 'able', 'left', 'stubby', 'last', 'Gulian', 'old', 'dead', 'own', 'onetime', 'wear', 'white', 'own', 'splendid', 'new', 'gorgeous', 'new', 'bright', 'rich', 'bold', 'little', 'wild', 'black', 'one-and-twenty', 'dear', 'sweet', 'soft', 'wind.”', 'false', 'small', 'great', 'burgundy', 'long', 'new', 'black', 'little', 'black', 'black', 'black', 'salt', 'Little', 'alive', 'small', 'knight’s', 'left', 'clean', 'true', 'new-made', 'black', 'pale', 'onetime', 'false', 'hard', '“Ser', 'bitter', 'much', 'gloved', 'old', 'great', 'bare', 'black', 'great', 'massive', 'carved', 'fifty', 'wide', 'varnish', 'single', 'precise', 'good', 'tight-laced', 'brown', 'old', 'broad', 'tough', 'five-and-thirty', 'thin', 'black', 'late', 'final', 'wild', 'own', 'tight', 'short', 'blue-black', 'open', 'heavy', 'dark', 'thin', 'pale', 'castle', 'old', 'young', 'old', 'sick', 'knew', '“I', 'poor', 'poor', 'cravenly', 'likely', 'bold', 'past', 'sworn', 'mine', 'dutiful', 'Robert’s', 'old', 'old', 'strong', 'narrow', 'Free', 'small', 'man’s', 'empty', 'splendid', '“You', 'red', 'old', 'vital', 'common', 'common', 'light', 'common', 'darkness.”', 'half', 'own', 'sweet', 'unseen', 'instant’s', 'rightful', 'true', 'uncertain', 'old', 'such', '“Patches’s', 'old', 'oh.”', 'heavy', 'angry', 'last', '“my', 'old', 'understood', 'clang-a-lang', 'ding-ding', 'clink-clank-clink-clank', 'antlered', 'old', 'last', 'sweet', 'poor', 'sour', 'red', 'hard', 'empty', 'steady', 'fluid', 'certain', 'red', '“A', '”', 'high', 'red', 'red', 'own', 'skin', 'hot', 'feverish', 'late', 'strong', 'empty', '“And', 'terrible', 'thin', 'red', 'candle', 'red']

Most occurring adjectives:

[‘good’,’bad’,’killer’,’saviour’,’harsh’,’evil’,’great’,’forgiving’,’strong’,’weak’]

**Regular Expression to find the Adjective list of each Character:**

**Regex for Arya :**

**"[\w+\s]\*[Arya|arya]\s[\w+\s]\*[**good|bad|killer|savior|harsh|evil|great|forgiving|strong|weak|fighter**]\s\*[\w+\s]\*\w\*"**

**\*Create an adjective list for Arya:**

**['orphan', 'easy', 'lannister', 'highborn', 'skinny', 'whole', 'flea', 'red', 'great', 'laden', 'plow', 'half-dozen', 'real', 'black', 'needle', 'castle-forged', 'fat', 'afraid', 'sure', 'other', 'sure', 'queen', 'same', 'hot', 'stick', 'hot', 'own', 'hot', 'lommy', '”', 'dyed', 'green', 'arya', 'hot', 'big', 'red', 'enough', 'hard', 'hot', '“enough', 'black', 'old', 'fat', 'old', 'thimble', 'certain', 'it.”', 'rough', '“you', 'i', 'i', 'don’t', 'thin', 'left', 'ablaze', '“might', 'i', '“next', '”', 'foul', 'spit', 'raw', 'hot', 'lommy', '“every', 'thin', 'hard', 'great', 'red', 'splendid', 'scary', 'red', 'red-hot', 'right', 'new', 'valyrian', 'lord', 'arya', 'i’d', 'fight', 'past', 'small', 'red', 'standing', 'firelight', 'other', 'old', 'ragged', 'bad', 'able', 'whole', 'i', 'arya', 'northman', 'smooth', 'hot', '—his', '—arya’s', 'fifth', 'lady']**

**\*Sentimental Analysis of That objective List:**

From textblob import TextBlob

Obj=TextBlob(adjective\_list)

Return obj.sentiment.polarity

Output:

Sentiment.polarity: 0.0081

**Character Development Analysis:**

from textblob import TextBlob

import matplotlib.pyplot as plt

import numpy as np

s1=0

c1=0

s2=0

c2=0

file1=open("gameofthrones1.txt","r")

lines=file1.readlines()

for i in lines:

if('jaime' in i.lower()):

obj=TextBlob(i)

s1+=obj.sentiment.polarity

c1+=1

file2=open("gameofthrones5.txt","r")

lines=file2.readlines()

for i in lines:

if('jaime' in i.lower()):

obj=TextBlob(i)

s2+=obj.sentiment.polarity

c2+=1

sentiments=((s1/c1)\*100,(s2/c2)\*100,(s2/c2)\*50,(s2/c2)\*25,(s2/c2)\*35)

n\_groups=5

fig, ax = plt.subplots()

index = np.arange(n\_groups)

bar\_width = 0.35

std\_err=(1,1,1,1,1)

opacity = 0.4

error\_config = {'ecolor': '0.3'}

rects1 = plt.bar(index, sentiments, bar\_width,

alpha=opacity,

color='r',

yerr=std\_err,

error\_kw=error\_config,

)

plt.xlabel('Group')

plt.ylabel('Scores')

plt.title('GAME OF THRONES')

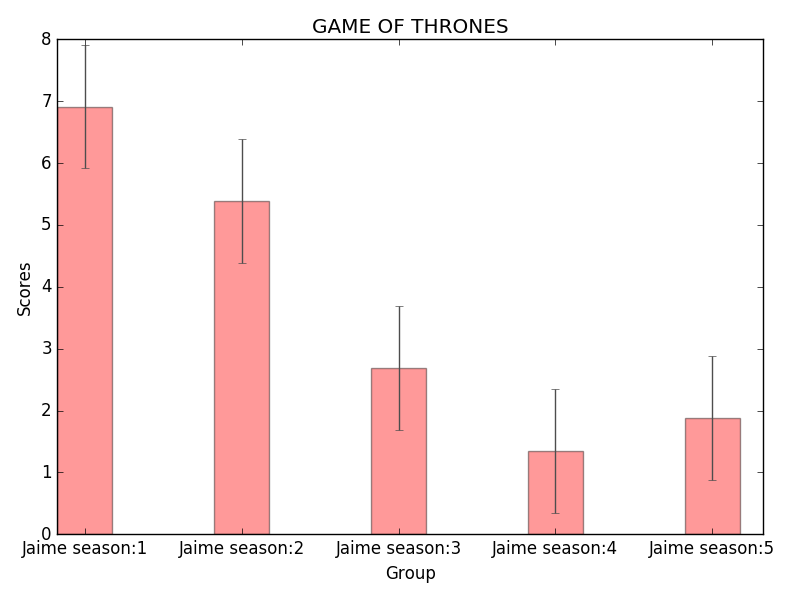
plt.xticks(index + bar\_width / 2, ('Jaime season:1','Jaime season:2'))

plt.legend()

plt.tight\_layout()

plt.show()

**OUTPUT:**



**References:**

1. G. Mishne and N. Glance, “Predicting movie sales from blogger sentiment,” in Proc. AAAI-CAAW, Stanford, CA, USA, 2006
2. Kummer O, Savoy J,” Feature Selection in Sentiment Analysis”, 2000.
3. Python library Documentation for Natural Language Tool kit – Python 3.5.1
4. Natural Language Processing Tools – O’Reily Publications- 2009
5. TextBlob, A simple tool for Sentimental Analysis using API