Configuration Manual

MSc Research Project
Data Analytics

Aditya Kumar Singh

Student ID: x20140410

School of Computing
National College of Ireland

Supervisor: Noel Cosgrave

Student Name:	Aditya Kumar Singh
Student ID:	x20140410
Programme:	Data Analytics
Year:	2021
Module:	MSc Research Project
Supervisor:	Noel Cosgrave
Submission Due Date:	31/1/2022
Project Title:	Configuration Manual
Word Count:	XXX
Page Count:	5

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Aditya Kumar Singh
Date:	30th January 2022

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

Aditya Kumar Singh
x20140410

1 Introduction

The aim of the research project is to determine the effectiveness of sentiment analysis in DOTA 2 videogame and how it can be used to identify cyberbullying after incorporating specific word and nuances common to the game. This configuration manual is a guide for the replication of the project for all stages. Several code snippets are provided for successful implementation of the reproduction.

2 System Configuration

Table 1: System Configuration

	Specification
Operating System	Windows 10 Home Single Language 64-bit
Memory	16.0 GB RAM
CPU	AMD Ryzen 7 4800H
Cores	16
GPU	AMD Radeon RX 5600M Series
Storage	500 GB SSD

Environment: Anaconda Navigator 2.1.1
Programming language: Python 3.8.12
IDE: Jupyter Notebook

3 Data Selection

The datasets for the project can be download from the following URL links:

1. **DOTA 2 chat Dataset:** `https://www.kaggle.com/romovpa/gosuai-dota-2-game-chats?select=dota2_chat_messages.csv`

2. **Bing Lexicon Dictionary:** `https://www.kaggle.com/andradaolteanu/bing-nrc-afinn-lexicons?select=Bing.csv`

3. **Positive sentiment dataset** `https://github.com/MiyainNYC/Text-Sentiment-Analysis-/tree/master/pos`

4. **Negative sentiment dataset** `https://github.com/MiyainNYC/Text-Sentiment-Analysis-/tree/master/neg`

Table 2: Python Libraries and their version

Library	version
pandas	1.3.4
numpy	1.21.2
glob2	0.7
nltk	3.6.5
matplotlib	3.5.0
wordcloud	1.8.1
vaderSentiment	3.3.2
ipython	7.29.0
scikit-learn	1.0.1
jupyter	1.0.0
jupyter-client	6.1.12
jupyter-console	6.4.0
jupyter-core	4.9.1

4 Code Snippets

The project is divided into three python files. One is for DOTA 2 chat dataset, and other 2 are for Bing lexicon dictionary.

4.1 Libraries

For the sake of simplicity all the libraries have been assembled in one place and cun be run on all the files.

```
1  import pandas as pd
2  import numpy as np
3  import csv
4  import sys
5  import os
6  import glob
7  import nltk
8  import matplotlib.pylab as plt
9  import seaborn as sns
10 from wordcloud import WordCloud
11 from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer
12 from IPython.display import Image
13 from sklearn.metrics import confusion_matrix
14 from sklearn.metrics import classification_report
```

4.2 Loading Data

In this section loading data code is available.

4.2.1 Loading Data for DOTA 2 dataset

```
1  dfx = pd.read_csv ('dota2_chat_messages.csv')
```

change the directory to the same folder to the notebook directory or change the path to the downloaded directory

4.2.2 Opinion Dataset

This code imports and apends the same dataframe at the same time.

```
def isNotNull(value):
    return value is not None and len(value)>0

neg_file = []
os.chdir('C:\\the\\neg')
for file in glob.glob('*.txt'):
    neg_file.append(file)
neg_content = []
for i in range(len(neg_file)):
    txt = open(neg_file[i])
    neg_content.append(txt.read())

pos_file = []
os.chdir('C:\\the\\pos')
for file in glob.glob('*.txt'):
    pos_file.append(file)
pos_content = []
for i in range(len(pos_file)):
    txt = open(pos_file[i])
    pos_content.append(txt.read())
```

4.2.3 Vanilla Bing Lexicon Dataset

Change the directory path as usual. And the data will be appended.

```
os.chdir('C:\\the')
dict_pos = []
dict_neg = []
f = open('negative-words.txt','r')
for line in f:
    t= line.strip().lower();
    if (isNotNull(t)):
        dict_neg.append(t)
f.close()

f = open('positive-words.txt','r')
for line in f:
    t = line.strip().lower();
    if (isNotNull(t)):
        dict_pos.append(t)
f.close()
```

4.2.4 Updated Bing Lexicon Dataset

Change the directory path as usual.

```
os.chdir('C:\\the')
dict_pos = []
dict_neg = []
f = open('negative-words-updated.txt','r')
for line in f:
    t= line.strip().lower();
    if (isNotNull(t)):
        dict_neg.append(t)
```

```
 9  f.close()
10
11  f = open('positive-words-updated.txt','r')
12  for line in f:
13      t = line.strip().lower();
14      if (isNotNull(t)):
15          dict_pos.append(t)
16  f.close()
```

4.3 Model Implementation

4.3.1 VADER

Run the model

```
1  analyser = SentimentIntensityAnalyzer()
2  def sentiment_analyzer_scores(sentence):
3      score = analyser.polarity_scores(sentence)
4      print("{:-<40} {}".format(sentence, str(score)))
```

4.3.2 Bing Lexicon

```
 1
 2  nalysis_for_pos = []
 3  for i in range(len(Bing_senti)):
 4      tokens = nltk.word_tokenize(pos_content[i])
 5      neg_cnt = 0
 6      pos_cnt = 0
 7      for neg in dict_neg:
 8          if (neg in tokens):
 9              neg_cnt = neg_cnt +1
10      for pos in dict_pos:
11          if (pos in tokens):
12              pos_cnt = pos_cnt +1
13      analysis_for_pos.append(pos_cnt - neg_cnt)
14  Bing_senti['Bing_analysis_for_pos'] = analysis_for_pos
15
16
17  analysis_for_neg = []
18  for i in range(len(Bing_senti)):
19      tokens = nltk.word_tokenize(neg_content[i])
20      neg_cnt = 0
21      pos_cnt = 0
22      for neg in dict_neg:
23          if (neg in tokens):
24              neg_cnt = neg_cnt +1
25      for pos in dict_pos:
26          if (pos in tokens):
27              pos_cnt = pos_cnt +1
28      analysis_for_neg.append(pos_cnt - neg_cnt)
29  Bing_senti['Bing_analysis_for_neg'] = analysis_for_neg
```

4.4 Results

4.4.1 VADER

Run this code to get polarity result. Put the term between quotes.

```
1  sentiment_analyzer_scores("gg")
2  sentiment_analyzer_scores("?")
3  sentiment_analyzer_scores("GGWP")
```

4.4.2 BING lexicon

this snippet will append the result for all opinion

```
1
2  neg_analysis_label = []
3  for i in analysis_for_neg:
4      if i >0:
5          neg_analysis_label.append(1)
6      else:
7          neg_analysis_label.append(0)
8
9  pos_analysis_label = []
10 for i in analysis_for_pos:
11     if i >0:
12         pos_analysis_label.append(1)
13     else:
14         pos_analysis_label.append(0)
```

For tabular results, this code is needed.

```
1  Bing_senti['analysis_label_for_neg'] = neg_analysis_label
2  Bing_senti['analysis_label_for_pos'] = pos_analysis_label
3  Bing_senti['label_for_neg'] = [0]*len(Bing_senti)
4  Bing_senti['label_for_pos'] = [1]*len(Bing_senti)
5  Bing_senti.head()
```

4.5 Evaluation

This code will gove all the evaluation parametrs for the Bing lexicon

```
1  classification_report(True_label,Bing_analysis)
```