National College of Ireland

Configuration Manual

MSc Research Project
Data Analytics

Shrinidhi Chandrashekhar Shetty
X18199780@student.ncirl.ie

School of Computing
National College of Ireland

Supervisor: Vladimir Milosavljevic

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Shrinidhi Chandrashekhar Shetty
Student ID:	18199780
Programme:	MSc. Data Analytics **Year:** 2019-20
Module:	MSc. Research Project
Lecturer:	Vladimir Milosavljevic
Submission Due Date:	17-August-2020
Project Title:	A Deep Learning Approach for Suicide Risk Assessment using Reddit

Word Count: 900 **Page Count:** 20

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: Shrinidhi C Shetty

Date: 16-August-2020

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

Shrinidhi Chandrashekhar Shetty
Student ID: x18199780@student.ncirl.ie

1 Requirements

The hardware, software, libraries required to conduct the research experiment is as follows:

1.1 Hardware Requirements

Operating System: Windows 10
Processor: Intel Core i3 or above
Memory: 8GB
Hard Disk Drive / Solid State Drive: 20GB
Hardware Accelerator: GPU Inbuilt in Google Collaboratory (Nvidia K80s, T4s, P4s, P100s)

1.2 Software Requirements

IDE: Google Collaboratory [Cloud-based Jupyter Notebook]
Python: 3.6
API: pushshift.io
Documentation Tools: Microsoft Office, Microsoft Excel, Microsoft PowerPoint.
Web Browser: Google Chrome

1.3 Packages Required

Pandas
Numpy
Matplotlib
Seaborn
Itertools
NLTK
Scikit-learn
spaCy
Keras
Torch
Regular Expression

Installation command: pip install <package-name>

1.4 Word- Embedding Required

Global Vectors (GloVe)

2 Environment Setup

Sign in to Google Collaboratory with Gmail credentials [1] and Mount Google Drive to access all the data sources [2] as shown in figure 1.

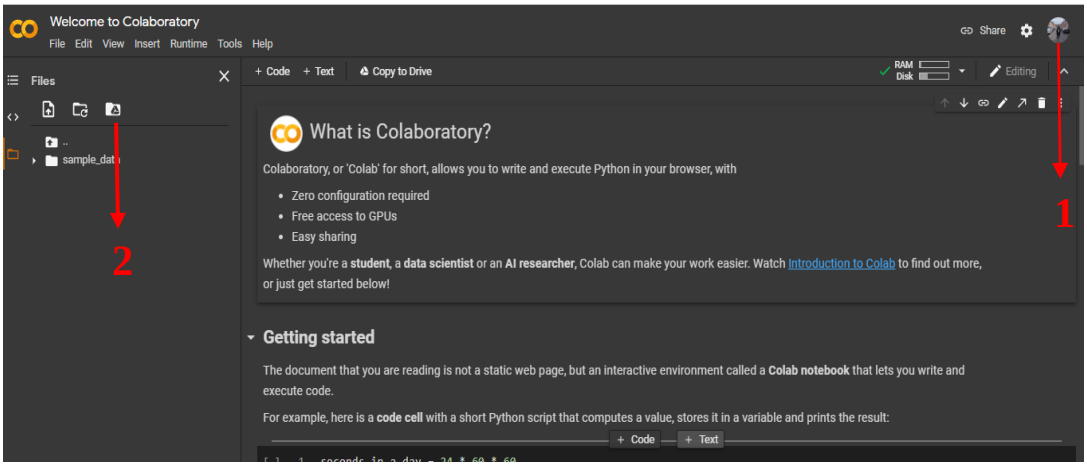

Figure 1: Google Collaboratory Setup

3 Data Sources

The libraries for each of the Code File are listed in the first cell of the file. They will be imported accordingly. All the data files are stored separately at the end of each python file.

Research contains two data sources, namely labeled dataset, and unlabelled dataset.

- **Labelled Dataset**: The dataset[1] is built based on the medical lexicons. They are extracted from Reddit and are annotated manually with domain expertise. (Gaur et al, 2019).
- **Unlabelled Dataset**: The data is scraped from Reddit – 'r/SuicideWatch'. API used to scrap is pushshift.io[2]. The code snippet for data scraping is shown in Figure 2.

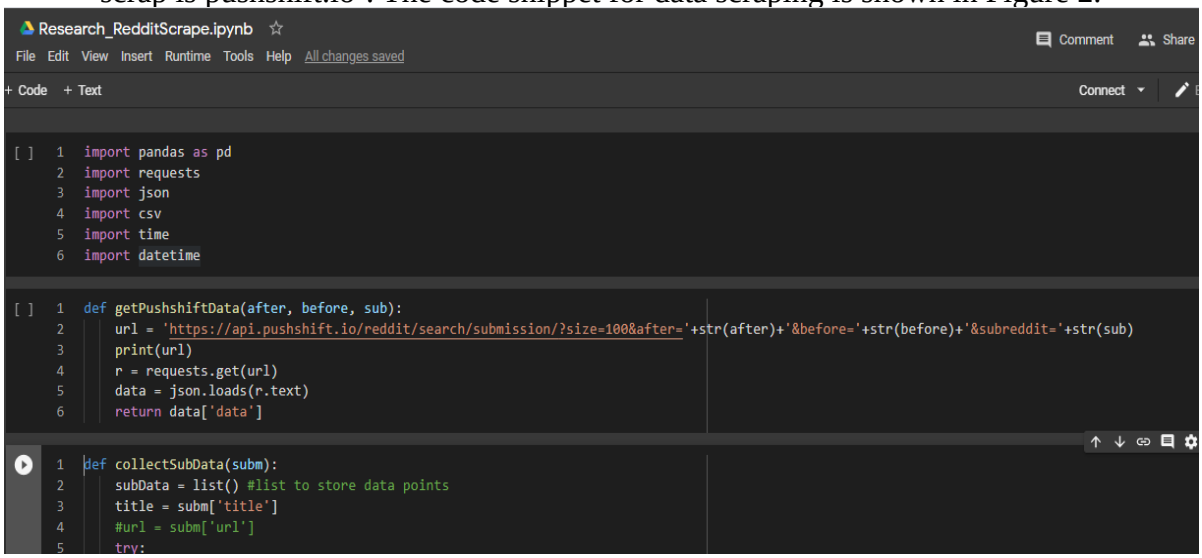

Figure 2: Data Scraping from Reddit using pushshift.io

[1] https://github.com/AmanuelF/Suicide-Risk-Assessment-using-Reddit
[2] https://pushshift.io/api-parameters/

Labeled dataset: Reddit_Raw_Data.xlsx
Unlabeled dataset: Unlabelled_Data.xlsx

4 Data Cleaning

The data sources are cleaned for removing outliers, missing values, noisy data, and form structures textual data. The code snippet for Data Cleaning is shown in Figure 3. The main package for the cleaning process is the spaCy[3] library. This step is conducted on the labeled dataset and unlabelled dataset separately.

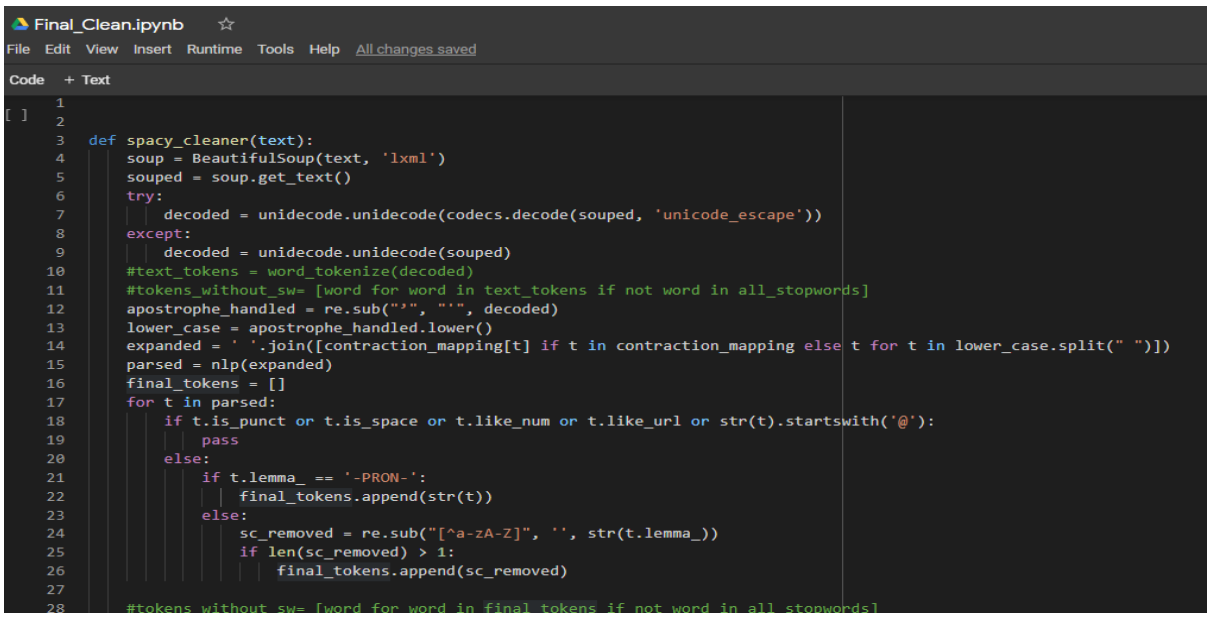

Figure 3: Data Cleaning

Labeled dataset: Clean_Reddit_Raw_Data.xlsx
Unlabeled dataset: Unlabelled_Data_Clean.xlsx

5 Data Augmentation

Enhancing the diversity of the initial cleaned dataset takes place here. This step produces a well-balanced dataset. The code snippet is shown in Figure 4. This step is processed on a cleaned labeled dataset from the previous step.

[3] https://pypi.org/project/spacy/

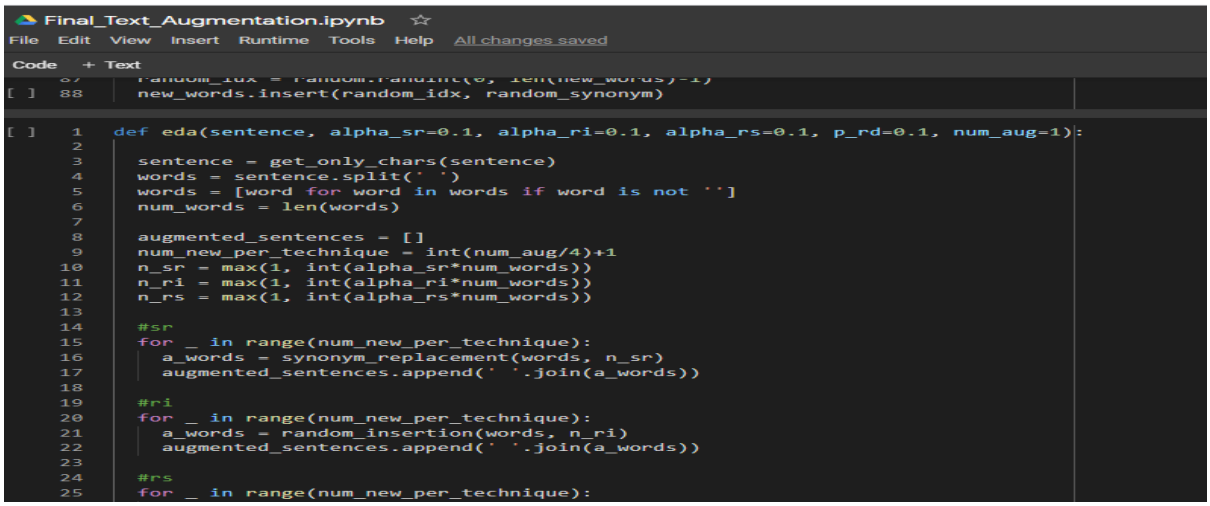

Figure 4: Data Augmentation

Labeled dataset: Aug_Clean_Reddit_Raw_Data.xlsx

6 Semi-Supervised Learning

This step is processed using the cleaned unlabelled dataset scraped from Reddit. The code snippet is shown in Figure 5.

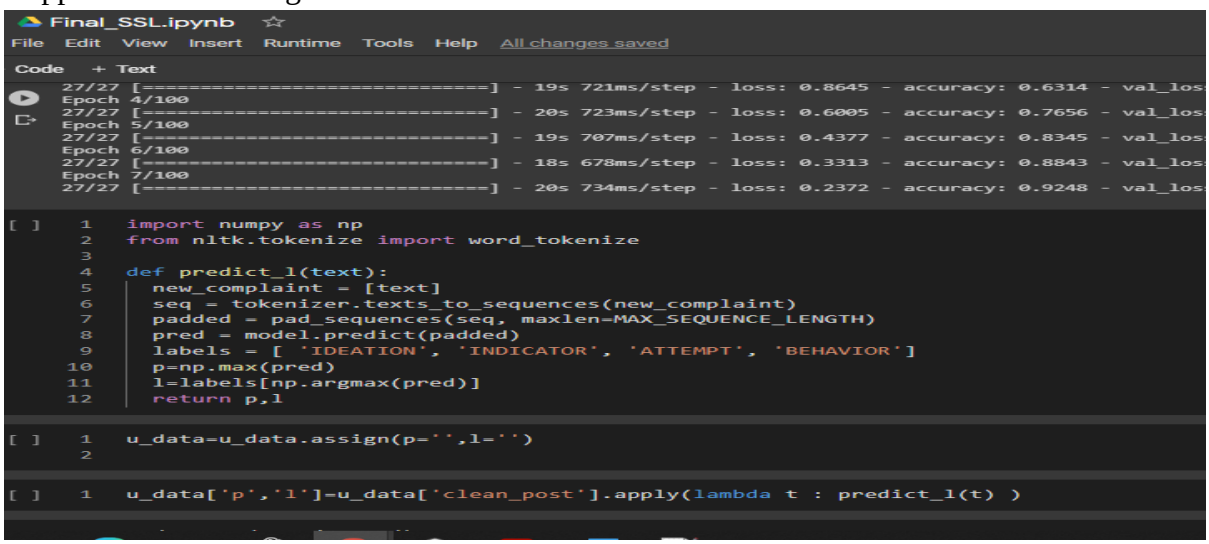

Figure 5: Self-Learning

Unlabelled dataset: semipred_data.csv

7 Data Modelling

The files Aug_Clean_Reddit_Raw_Data.xlsx and semipred_data.csv are merged and algorithms are applied. This part consists of two models namely Text CNN and BiLSTM.

- Hardware Accelerator Set up: GPU must be selected in the Notebook settings before running the code. The snapshot of the same is shown in Figure 6.

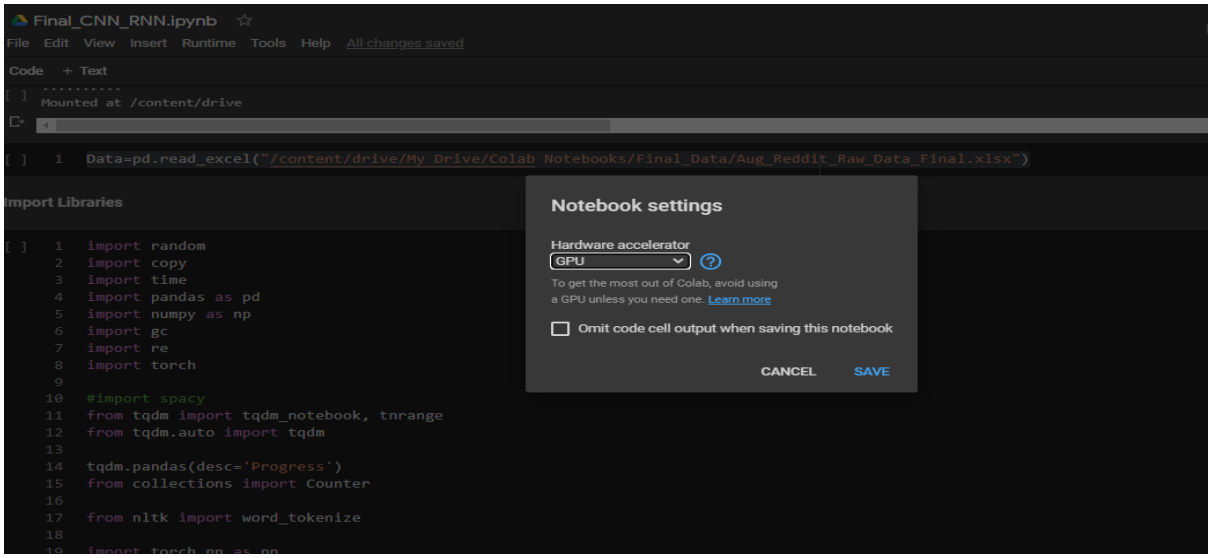

Figure 6: Google Collab Notebook Settings

- Loading GloVe embeddings: The pre-trained word embeddings called GloVe must be downloaded[4] and inserted in Google drive or the code to directly download in the Google Collab is written. The snapshot of the same is shown in Figure 7.

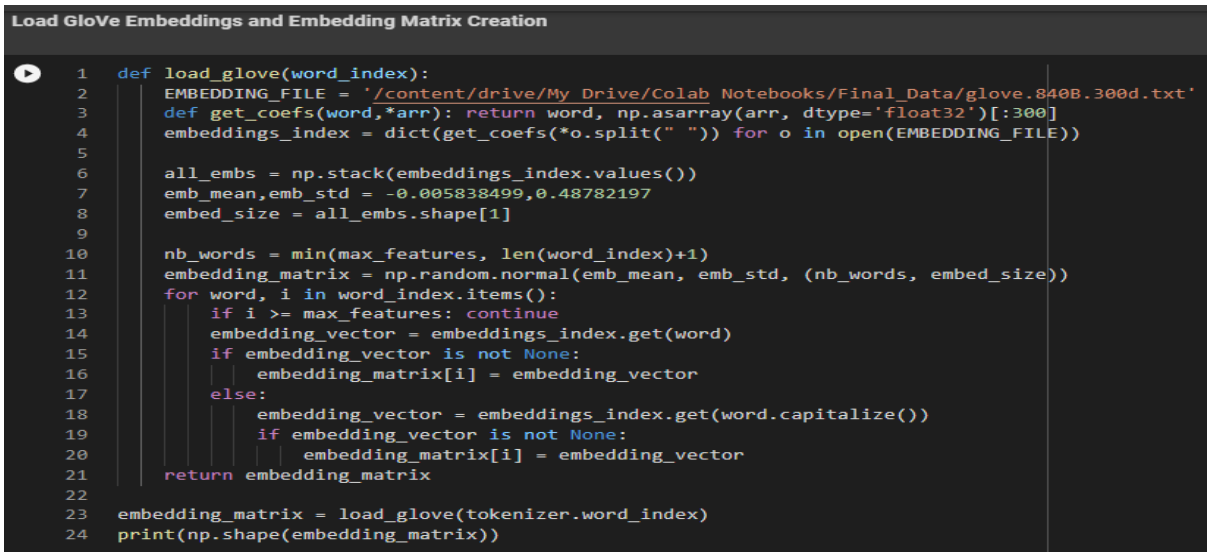

Figure 7: GloVe Embeddings

- Text Convolution Neural Network (TextCNN)
  The code snippet for the Text CNN model is shown in Figure 8.

[4] http://nlp.stanford.edu/data/glove.840B.300d.zip

TextCNN Algorithm

```
[ ]  1  class CNN_Text(nn.Module):
     2
     3      def __init__(self):
     4          super(CNN_Text, self).__init__()
     5          filter_sizes = [1,2,3,5]
     6          num_filters = 36
     7          n_classes = len(le.classes_)
     8          self.embedding = nn.Embedding(max_features, embed_size)
     9          self.embedding.weight = nn.Parameter(torch.tensor(embedding_matrix, dtype=torch.float32))
    10          self.embedding.weight.requires_grad = False
    11          self.convs1 = nn.ModuleList([nn.Conv2d(1, num_filters, (K, embed_size)) for K in filter_sizes])
    12          self.dropout = nn.Dropout(0.1)
    13          self.fc1 = nn.Linear(len(filter_sizes)*num_filters, n_classes)
    14
    15
    16      def forward(self, x):
    17          x = self.embedding(x)
    18          x = x.unsqueeze(1)
    19          x = [F.relu(conv(x)).squeeze(3) for conv in self.convs1]
    20          x = [F.max_pool1d(i, i.size(2)).squeeze(2) for i in x]
    21          x = torch.cat(x, 1)
    22          x = self.dropout(x)
    23          logit = self.fc1(x)
    24          return logit
```

Figure 8: Text CNN

- Bidirectional Long Short Term Memory(BiLSTM)
  The code snippet for BiLSTM Model is shown in Figure 9.

BiLSTM Algorithm

```
 1  class BiLSTM(nn.Module):
 2
 3      def __init__(self):
 4          super(BiLSTM, self).__init__()
 5          self.hidden_size = 64
 6          drp = 0.1
 7          n_classes = len(le.classes_)
 8          self.embedding = nn.Embedding(max_features, embed_size)
 9          self.embedding.weight = nn.Parameter(torch.tensor(embedding_matrix, dtype=torch.float32))
10          self.embedding.weight.requires_grad = False
11          self.lstm = nn.LSTM(embed_size, self.hidden_size, bidirectional=True, batch_first=True)
12          self.linear = nn.Linear(self.hidden_size*4 , 64)
13          self.relu = nn.ReLU()
14          self.dropout = nn.Dropout(drp)
15          self.out = nn.Linear(64, n_classes)
16
17
18      def forward(self, x):
19          #rint(x.size())
20          h_embedding = self.embedding(x)
21          #_embedding = torch.squeeze(torch.unsqueeze(h_embedding, 0))
22          h_lstm, _ = self.lstm(h_embedding)
23          avg_pool = torch.mean(h_lstm, 1)
24          max_pool, _ = torch.max(h_lstm, 1)
```

Figure 9: BiLSTM

8 Data Visualisations

The final code is visualizations of the text corpus for interpretation purposes. The code snippet is shown in Figure 10.

Figure 10: Data Visualizations

References

Manas Gaur, Amanuel Alambo, Joy Prakash Sain, Ugur Kursuncu, Krishnaprasad Thirunarayan, Ramakanth Kavuluru, Amit Sheth, Randy Welton, and Jyotishman Pathak. 2019. Knowledge-aware Assessment of Severity of Suicide Risk for Early Intervention. The World Wide Web Conference (WWW '19). Association for Computing Machinery, New York, NY, USA, 514–525. DOI:https://doi.org/10.1145/3308558.3313698