SCSE20-0985

AI for Healthcare

Submitted By: Neo Shun Xian Nicholas (U1820539F)

Project Supervisor: Associate Professor Erik Cambria

Project Examiner: Associate Professor Anupam Chattopadhyay

School of Computer Science and Engineering

Submitted in Partial Fulfilment of the Requirements for the Degree of Bachelor (Computer Engineering) of the Nanyang Technological University

2021

ABSTRACT

Mental health has been an increasingly challenging issue to tackle in this era due to the stressful environment we are living in. One such example of mental health illness is depression. Depression is a mood disorder described as feelings of sadness, loss or anger that interfere with one’s everyday activities. Depression has existed as a problem in this society for many years.

With technological advancements, social media platforms serve as a place for depressed personnel to seek help, hoping to feel better in one way or another. However, their problems are often neglected by others on the internet. If not detected quickly and accurately, one’s depression may develop into more serious issues such as suicidal thoughts.

Research has shown that nearly 300 million people in the world suffer from depression every year. Measures to assess depression include clinical judgement or structured interviews, but a more common method is the use of social media analysis. Social media helps to detect depression by analysing posts on social media platforms. This method is preferred as expressing one’s feelings online has become the new norm, and processing of social media data can take place quickly, so authorities are able to intervene at an earlier stage.

This project thus aims to analyse depressive texts from social media such as Twitter and Reddit by building various deep learning models for the different main tasks, hoping that we can detect depression and the cause of depression at an earlier stage. These main tasks include Classification, Emotion Intensity Prediction and Emotion-Cause Pair Extraction.

ACKNOWLEDGEMENTS

I would like to take this opportunity to express my utmost gratitude towards my project supervisor, **Associate Professor Erik Cambria**, for his guidance, his helpful suggestions and also his valuable advice during the project despite his busy schedule. His rich experience in Natural Language Processing and Sentiment Analysis have greatly benefitted me in completing this project as I am more exposed to a few of the current research areas, one of which is the Emotion-Cause Pair Extraction.

I would also like to thank my examiner, **Associate Professor Anupam Chattopadhyay,** for grading this Final Year Project.

I am also thankful towards the **Sentic Team**, for providing various Application Programming Interfaces (APIs) to assist in my experimentation and research in this project.

In addition, I would like to express my appreciation to **Shaoxiong Ji,** a Doctoral Candidate in Aalto University, Finland, for suggesting sites to obtain text datasets that are related to depression which are deemed suitable for my research.

Lastly, I would like to thank my family and friends who have consistently supported and encouraged me throughout the project.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1: INTRODUCTION

1.1 Background and Motivation

Analysing text contents from social media platforms, such as Twitter and Reddit, has become a popular way to make predictions about human emotions in the field of research. With the wide population using these social media platforms, there exists a huge source of data to analyse human emotions [1].

An area of interest in analysing human emotions is the detection of depressive text contents because depression has been a pressing issue that affects a large part of the population worldwide [2] [3]. This brings the motivation to dive deeper into analysing depressive texts with a variety of Natural Language Processing (NLP) tasks, which will be discussed in section 1.4 of this report. Hopefully, these analyses will be useful for early detection of depression and the cause of depression. Hence, relevant authorities are able to take actions to curb one's depression at an earlier stage [4].

1.2 Challenges of Detecting Sentiments of Depression from Text

1.2.1 Sentiment Analysis vs Emotion Analysis

While often being used interchangeably, Sentiment Analysis and Emotion Analysis are quite different in terms of how unstructured text data are being analysed. Several industry experts see Emotion Analysis as the evolved form of Sentiment Analysis. Sentiment Analysis is limited by separating the data points and determining whether the text conveys a negative or positive feeling. In contrast, Emotion Analysis consists of deeper analysis that dives down into the psychology of one's emotion and behaviour [5].

1.2.2 Why Detecting Depression from Text is Hard?

Detecting depression from text data is an Emotion Analysis task rather than a Sentiment Analysis task because the labels "depressed/not depressed" cannot be equated with the labels "positive/negative". In addition, indications of depression are usually subtle and not obvious to the readers unless they are from specific fields that analyse emotions and text such as the Psychologists or the Linguists. These subtle indications may be reflected in the nuances of someone's language [6]. Hence, a more in-depth psychological analysis should be conducted to assess the text data.

1.3 Emergence of Deep Learning Solutions

Traditionally, a rule-based method is used for sentiment or emotion analysis tasks, where the algorithm calculates the sentiment score from a set of manually created rules. For example, one can have a rule to count the number of times the word "depressed", "suicide" or "sad" appear in a particular social media text, and increase the estimated sentiment to decide if this piece of text contains sentiment of depression or not [7]. However, a rule-based system requires a deep knowledge of the domain and the need to manually create many rules for the best performance, making this approach very time consuming and not scalable for larger datasets.

With the huge amount of available social media data, deep neural network (DNN) models have seen significant success in the performance of NLP-related tasks. There are a variety of DNN models proposed for sentiment and emotion analysis, such as the Recurrent Neural Network (RNN) and also Convolution Neural Network (CNN).

In addition, the introduction to the state-of-the-art Transformer models enhanced the performance of NLP-related tasks even further as stated in the "Attention is All You Need" research paper [8]. Like the RNN, a Transformer is an architecture for transforming one sequence into another one with the help of the encoder and the decoder. It differs from RNN in that it does not imply any recurrent networks [9]. The main advantage of Transformer over RNN models is that they are not sequential, implying that they can be more easily parallelised. Hence, Transformer models can be easily scaled up as the models can be trained by parallelising the training [10].

1.4 Project Objective & Scopes

The focus of this project will be to leverage on the use of Deep Learning and NLP to do an in-depth analysis on depressive social media contents. Specifically, we will begin with the more common task of predicting whether a particular text contains sentiment of depression. Then we will dive deeper to create a depression metric and predict depression scores for all of the data that are labelled as depressive content. Lastly, we will experiment with the more uncommon task of

Emotion-cause Pair Extraction, which is to determine how accurate the model can extract the emotion and the cause of depression from text clauses.

1.5 Tasks in this Project

In this project, there will be a total of three main tasks and one subtask to perform an in-depth analysis on depressive social media texts. These tasks are stated below:

Main Tasks:

1. Emotion Classification
2. Emotion Intensity Prediction
3. Emotion-cause Pair Extraction

Subtask:

1. Text Summarisation (Extractive and Abstractive)

1.6 Report Organisation

The report is organised as follows:

- Chapter 1: Introduction
- Chapter 2: Literature Review
- Chapter 3: Project Setup
- Chapter 4: The Datasets
- Chapter 5: API and Libraries used in this Project
- Chapter 6: Main Task 1 - Emotion Classification
- Chapter 7: Main Task 2 - Emotion Intensity Prediction
- Chapter 8: Subtask - Text Summarisation
- Chapter 9: Main Task 3 - Emotion-cause Pair Extraction
- Chapter 10: Conclusion and Future Works

CHAPTER 2: LITERATURE REVIEW

This chapter explains the important terms and concepts that will be used in later chapters.

2.1 Techniques

This subsection gives a thorough explanation of the techniques used throughout this whole project.

2.1.1 Natural Language Processing (NLP)

NLP is a subset technique of Artificial Intelligence (AI) which is used to narrow the communication gap between the computer and the human [11]. It is a research field that enables the computer to think, extract and leverage on human language data. The ultimate goal for NLP researchers is to make use of human language data and develop computational models for machines to mimic the capabilities of a human such as speaking, writing, reading and listening [12]. NLP datasets can be in the form of text, audio or speech data.

NLP can be further broken down into two sub-categories, they are Natural Language Generation (NLG) and Natural Language Understanding (NLU). NLG is the process of writing or generating language and it is the component that helps to generate natural language using machines. NLG starts from facts such as Part-of-Speech (POS) tags and parsing results to generate the natural language. In contrast, NLU is the process of reading and interpreting language which helps to explain the meaning behind the natural language. This component generates facts from natural language using tools such as POS tagger or parsers to develop NLP applications [13]. Some of the NLP tasks are: Information Retrieval, Sentiment/Emotion Analysis, Neural Machine Translation, Question Generation, Dialogue Understanding, Text-to-Speech, Speech-to-Text etc.

NLP has seen evolving from the traditional symbolic approaches, to the statistical approaches and later on to the connectionist approaches.

The symbolic approaches use first-order logic to perform deep analysis of linguistic phenomena and are based on well-understood knowledge representation schemes and associated algorithms. The rules in this rule-based method are manually created by experts in the field of Linguistics so that the computer systems can follow. This method requires a lot of domain knowledge in the field of interest because this knowledge will be converted to facts and rules in the rule-based system. As such, a system usually uses symbolic approaches if there is a presence of human-developed rules and lexicons. Typical application areas that use the symbolic approaches are information extraction, text categorisation, ambiguity resolution and lexical acquisition [14].

The statistical approaches took the probabilistic means, where often it uses huge corpora to build generalised models of linguistic phenomena based on actual examples of these phenomena provided by the text data without adding significant linguistic or domain knowledge. In other words, there are no rules created for these approaches, but rather they use observable data as the primary source of evidence and the ‘rules’ are determined based on those data using statistical means. Typical application areas that use the statistical approaches are speech recognition, lexical acquisition, parsing, part-of-speech tagging, collocations, statistical machine translation, statistical grammar learning, and so on [14].

Similar to statistical approaches, the connectionist approaches also build generalised models of linguistic phenomena. However, the difference here is that connectionist models combine statistical learning with various theories of representation. As such, these approaches are more flexible to transformation, inference and manipulation of logic formulae. These models are harder to observe because their architectures are less constrained than statistical ones. Typical application areas that use connectionist approaches are word-sense disambiguation, syntactic parsing and associative retrieval [14].

In this project, we will mostly be looking at the connectionist approaches for the analysis, with a small portion of it using the symbolic approaches.

2.1.2 Deep Learning

Deep Learning is a subfield of machine learning concerned with algorithms that got its inspiration by the structure and the function of the brain called Artificial Neural Network (ANN). It is also a technique used to process data through many layers of non-linear transformations [15].

There are mainly two types of deep learning architecture in the supervised learning method, they are Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN). CNN is mainly used in Computer Vision, whereas RNN is suitable for sequential data and mainly used in NLP. While CNN can also be used in some of the NLP tasks in this project, the primary focus will be mainly focusing on RNN architectures as they are able to capture sequential dependencies in the analysed text.

2.2 Sequence Model Architectures

This subsection gives the definition of the sequence model architectures used in some of the chapters and also a brief comparison between one another.

2.2.1 RNN

An RNN is a type of ANN which uses sequential or time series data. RNN obtains information from prior inputs to influence the current input and output. Although traditional DNNs assume that inputs and outputs are independent of each other, the output of RNN is dependent on the prior elements within the sequence [16].

In general, there are five types of RNNs. Figure 1 to 5 below shows the different types of RNN and their applications [17][18].

One-to-one:

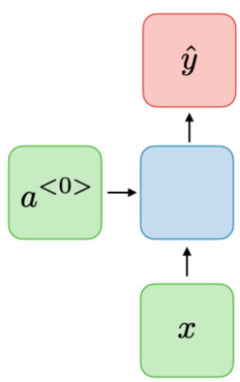

Figure 1: One-to-one RNN architecture

Figure 1 shows the classic feed-forward neural network architecture, with an input and an output.

One-to-many:

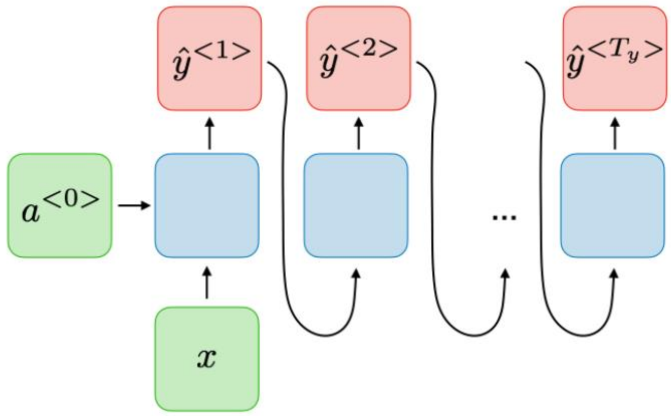

Figure 2: One-to-many RNN architecture

Figure 2 shows the RNN with fixed size input and the output can be variable in length. It is normally used in music generation or image captioning.

Many-to-one:

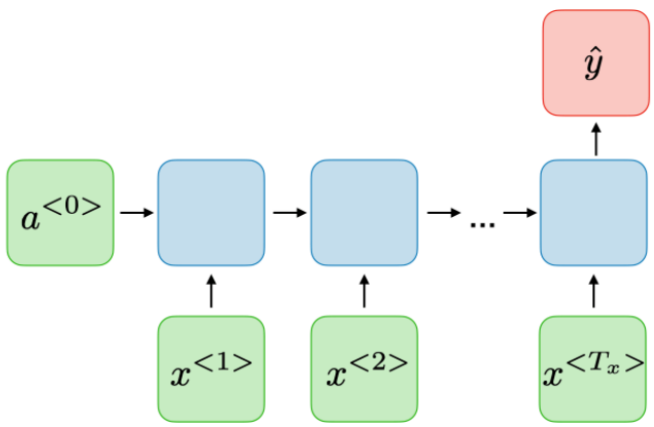

Figure 3: Many-to-one RNN architecture

Figure 3 shows the RNN that is normally used in sentiment classification. The input is expected to be a sequence of words or paragraphs. There will only be an output, which can be a regression output with continuous values which represents the likelihood of a sentence being positive/negative.

Many-to-many (T_x = T_y):

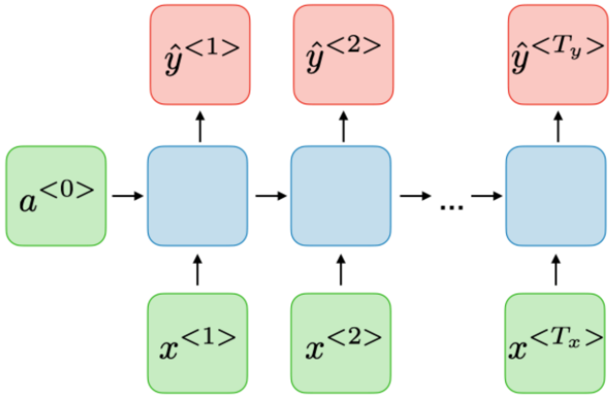

Figure 4: Many-to-many RNN architecture

Figure 4 shows the RNN that is normally used in name entity recognition.

Many-to-many ($T_x \neq T_y$):

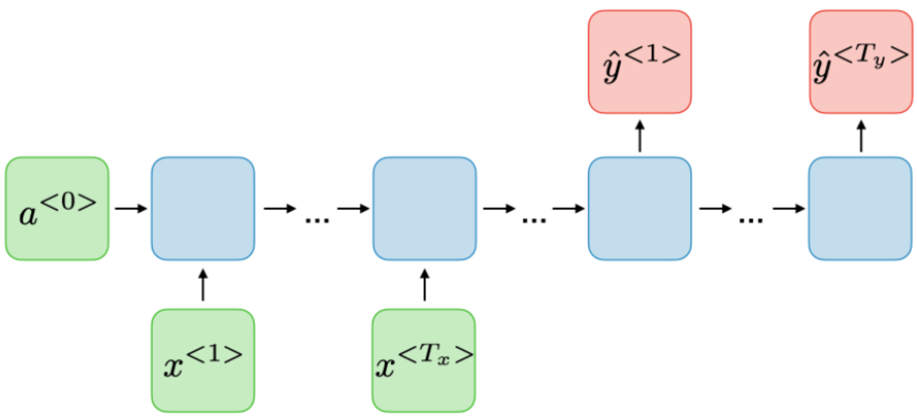

Figure 5: Many-to-many RNN architecture

Figure 5 shows the RNN that is normally used in machine translation. The input and output have a varying number of units in both many-to-many RNN architectures.

2.2.2 Limitations of vanilla RNNs

Despite the advancements of RNNs, there are some limitations. There exists the vanishing and exploding gradient problem and vanilla RNNs cannot be stacked up. Also, vanilla RNNs only have hidden states but not a cell state, which can be used to remove or add information to the cell. Thus, vanilla RNNs handle long-term dependency poorly as compared to the other variants of RNNs, which will be discussed in the next section.

2.2.3 Long Short-Term Memory (LSTM)

LSTM is one of the RNNs that overcomes the long-term dependency problem due to vanishing gradient problem. LSTM is dependent on three things, the current long-term memory of the network (cell state), the output at the previous point of time (hidden state) and the input data at the current time step. LSTM uses a series of gates to control the flow of the information [19]. Figure 6 shows a cell of the LSTM.

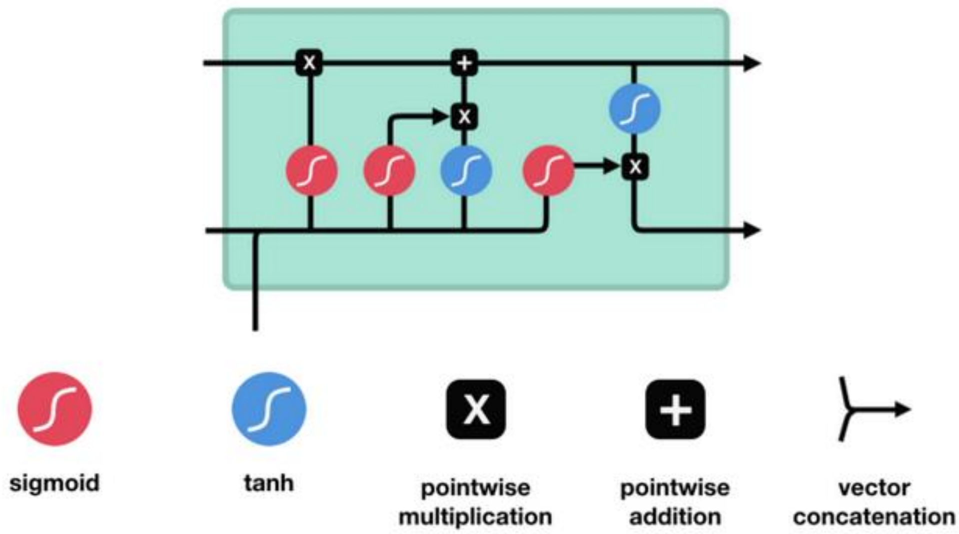

Figure 6: A cell in LSTM

The cell state transfers relative information throughout the sequence chain so that information from earlier time steps can make its way to later time steps, reducing the effort of short-term memory. As the cell state moves on, information gets added or removed to the cell state via gates. The gates are from another neural network that determines which information is permitted in the cell state. The gates can decide which information is relevant to keep or forget during training [19].

These gates contain sigmoid and tanh activations. Sigmoid activation squish values to between 0 and 1 whereas tanh activation squish values to between -1 and 1. Those are helpful to update or forget data as any number getting multiplied by zero will be zero, causing values to disappear or be "forgotten". In contrast, any number multiplied by one stays the same, or to be "kept" [19].

Forget Gate

This gate decides what information should be thrown away or kept. The sigmoid function is being fed with the previous hidden state and the current input information. Then, the output value from the sigmoid function will be between 0 and 1. If the value is close to 0, forget the information. If the value is close to 1, keep the information [19].

Input Gate

To update the cell state, the input gate is used. Firstly, the previous hidden state and current input is passed in a sigmoid function. The value after passing through the sigmoid function determines whether the value should be updated to 0 or 1. Also, the hidden state and current input will be passed to the tanh function to squish the values to between -1 and 1 to help regulate the network. Then, the tanh and sigmoid output will be multiplied together. The sigmoid output determines which information is important to keep from the tanh output [19].

Cell State

The cell state will first get pointwise multiplied by the forget vector. This has the possibility of dropping values in the cell state if it gets multiplied by values near zero. Then the output from the input gate will be pointwise added so as to update the cell state to new values that the neural network finds relevant, giving us the new cell state [19].

Output Gate

The output gate decides what the next hidden state should be. Firstly, the previous hidden state and the current input are passed into a sigmoid function. Then, the newly modified cell state will be passed to the tanh function. The tanh output and the sigmoid output will be multiplied with each other to decide what information the hidden state should carry. The output is the hidden state. Lastly, the new cell state and the new hidden state is then carried over to the next time step [19].

2.2.4 Gated Recurrent Unit (GRU)

GRU is the newer generation of RNN that is quite similar to an LSTM. GRU is less complex as compared to LSTM. GRU do not have the cell state and use the hidden state to transfer information. It only has two gates, a reset gate and update gate [19]. Figure 7 shows a cell of the GRU.

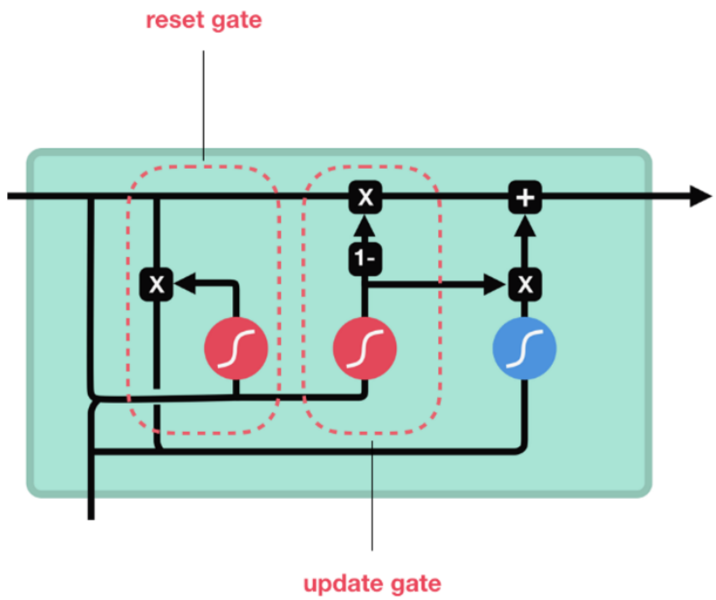

Figure 7: A cell in GRU

Update Gate

Similar to the forget gate and input gate of an LSTM, the update gate of a GRU decides what information to keep and what information to discard [19].

Reset Gate

The reset gate is another gate used to decide the amount of past information to discard [19].

2.3 Transformer Architecture

This subsection gives the definition of self-attention mechanism and the transformer model that will be used in this project.

2.3.1 Self-Attention and Transformers

With LSTM models, dealing with long-range dependencies is still challenging and the sequential nature of the model architecture prevents parallelisation. As such we will look at the Transformer architecture to solve these issues. Transformer aims to solve sequence-to-sequence tasks even with long-range dependencies. It is the model that relies on a self-attention mechanism to convert the input sequences into the output sequences without the need to be sequentially aligned such as the LSTM. Self-attention is the mechanism relating different positions of a single sequence in order to compute a representation of the sequence [20]. Figure 8 shows the Transformer's model architecture.

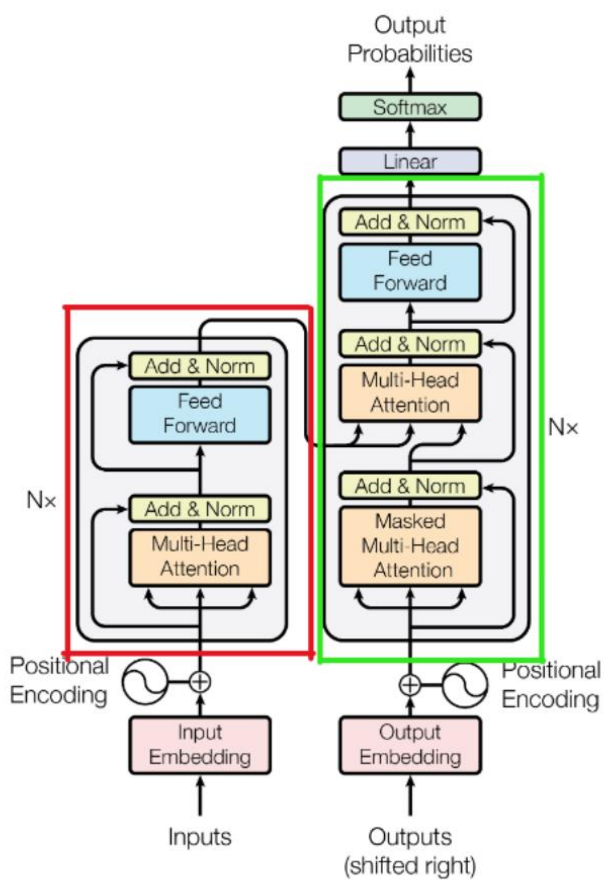

Figure 8: Transformer's model architecture

From Figure 8, the encoder portion is represented by the red box and the decoder portion is represented by the green box. The encoder and decoder blocks are actually multiple identical encoders and decoders stacked on top of each other and both have the same number of units. How the whole mechanism works is that the word embedding of the input sequence are first passed to the encoder and then propagate to the next encoder. The output of the last encoder stack is then passed to all the decoders in the decoder stack for decoding.

2.3.2 Bidirectional Encoder Representations from Transformers (BERT)

Specifically, BERT will be the main focus in this project when the implementation of the transformer is considered for some of the NLP tasks.

In relation to the state-of-the-art model in NLP, BERT combines two powerful technologies, it is based on a deep transformer network that is able to read in long texts by using attention and is also bidirectional, which takes into account the whole text passage to understand the meaning of each word. This is especially useful in this project's context because most of the depressive social media texts are long and descriptive [21]. Figure 9 shows the architecture of BERT.

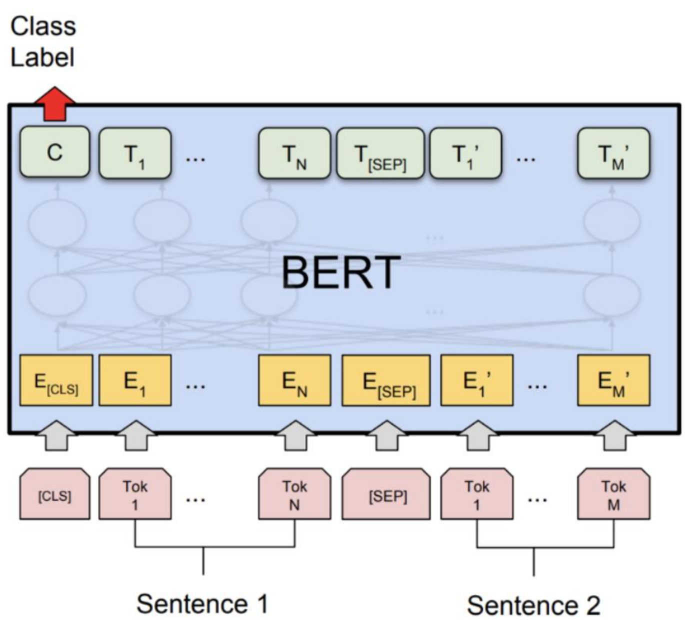

Figure 9: BERT's architecture

2.4 Word Embedding

In NLP, word embedding are vector representations of a particular word [22]. Commonly, it is a vector that encodes the meaning of the word in such a way that the words that are closer in the vector space are expected to be similar in meaning. Word embedding are obtained by the means of using a set of language modelling and feature learning techniques where words or phrases from the vocabulary are mapped to vectors of real numbers [23].

2.5 Hyperparameters

This subsection describes and explains the various hyperparameters that will be tuned in some of the tasks to achieve the most optimal performances.

2.5.1 Learning Rate

Learning rate is a hyperparameter that controls how much adjustments in the weights of the neural network with respect to the loss gradient. The smaller the value, the slower the descent towards the minimum. This may seem good using a small learning rate to ensure not to miss any local minima, however this implies more time is needed to converge, especially if stuck on a plateau region [24].

2.5.2 Batch Size

Batch size refers to the number of training examples utilised in one iteration [25]. There are three mode of batch size:

1. **Batch mode:** The batch size is equal to the total dataset, hence making the iteration and epoch values equivalent.
2. **Mini-batch mode:** The batch size is greater than one but less than the total dataset size. Normally, there will be a number that can be divided into the total dataset size.
3. **Stochastic mode:** The batch size is equal to one. Therefore, the gradient and the neural network parameters are updated after each sample.

In this project, mini-batch mode will be the default batch size option to be adopted.

2.5.3 Number of Epochs

The number of epochs is a hyperparameter that defines the number of times that the learning algorithm will iterate through the whole dataset [26].

2.5.4 Optimisers

Optimisers are algorithms used to change the attributes of the neural network such as weights and learning rate in order to reduce the losses [27]. Some commonly used optimisers are Stochastic Gradient Descent (SGD), Adaptive Moment Estimation (Adam) and Root Mean Square Propagation (RMSprop).

2.6 Performance Metrics

This subsection defines the performance metrics that are used in this project to compare the results of the different models trained with different hyperparameter values in some of the later chapters.

2.6.1 Accuracy

Accuracy measures how well the model predicts by comparing the predictions with the ground truth values in terms of percentage [28].

2.6.2 Loss

Loss is a value that sums up the errors in the trained model. It measures how well or bad a particular model is doing. If the errors are high, the loss will be high, implying that the model is not doing a good job. As such, the lower the loss, the better the model is doing [28].

For computing the loss, a loss function is used. Some common loss functions used are cross-entropy and mean squared error for classification and regression tasks respectively [28].

2.6.3 Precision

Precision is defined as the fraction of relevant instances among all retrieved instances [29].

2.6.4 Recall

Recall is the fraction of retrieved instances among all relevant instances [29].

2.6.5 F1-score

F1-score is a measure of a model's accuracy by combining the precision and recall of the model, and it is defined as the harmonic mean of the model's precision and recall [29]. The F1-score is defined as:

F1 = (2*Precision*Recall)/(Precision+Recall)

2.6.6 Confusion Matrix

Confusion matrix is a table that is used to describe the performance of a classification model on a set of test data for which the true values are known [30].

2.6.7 Area under the Curve (AUC)

AUC represents the degree of separability. It tells how much the model is capable of distinguishing between classes [31]. The higher the AUC, the better the model is predicting if the text data is depressive when the ground truth is actually depressive and predicting if the text data is non-depressive when the ground truth is actually non-depressive.

2.6.8 Receiver Operating Characteristics Curve (ROC Curve)

ROC is a probability curve that illustrates the diagnostic ability of a classifier system [32].

2.6.9 Inter-annotator Agreement

Inter-annotator Agreement is a measure of how well two or more annotators can make the same annotation decision for a certain text class [33].

2.6.10 Cohen's Kappa Coefficient

Cohen's Kappa Coefficient is a quantitative measure of the reliability for two raters that are rating the same text entries [34]. The equation for Cohen's Kappa Coefficient is:

$$\kappa = \frac{p_0 - p_e}{1 - p_e}$$

P_0 is the observed agreement and P_e is the expected agreement.

2.6.11 Mean Absolute Error (MAE)

MAE is calculated by the sum of the absolute value of error [35].

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i|$$

2.6.12 Mean Square Error (MSE)

MSE is an absolute measure of the goodness of fit.

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

MSE is calculated by the sum of squares of prediction error, which is the real output minus the predicted output and then divided by the number of data points [35].

2.6.13 Root Mean Square Error (RMSE)

RMSE is the square root of MSE.

$$RMSE = \sqrt{\sum_{i=1}^{n}\frac{(y_i - \hat{y_i})^2}{n}}$$

It is used more frequently than MSE because sometimes the MSE value can be too big to compare easily. Also, MSE is calculated by the square of error, hence the square root brings it back to the same level of prediction error, making it easier for interpretation [35].

2.6.14 Mean Absolute Percentage Error (MAPE)

MAPE is the average of the absolute percentage errors of forecasts. Error is defined as actual or observed value minus the forecasted value [35].

$$M = \frac{1}{n}\sum_{t=1}^{n}\left|\frac{A_t - F_t}{A_t}\right|$$

CHAPTER 3: PROJECT SETUP

3.1 Code Environment

The whole project will be coded in Python 3. Both the Jupyter Notebook and the Google Colaboratory will be used to implement the depression analysis on social media data.

3.1.1 Jupyter Notebook

Codes written in Jupyter notebooks are mainly dealing with importing of data and data cleaning for later analysis.

3.1.2 Google Colaboratory

Codes written in Google Colaboratory are mainly dealing with training of models for the different main and subtasks. Google Colaboratory is preferred for training of models because it has a free Graphics Processing Unit (GPU) that can be used to speed up training.

3.2 Libraries used in each task

The subsections below list down the libraries being used for each of the tasks.

3.2.1 Data Cleaning

- pandas
- numpy
- re
- os
- tqdm
- preprocessor
- cleantext
- autocorrect
- nltk
- sklearn

3.2.2 Emotion Classification

- pandas
- numpy
- os
- sys
- torch
- torchtext
- time
- sklearn
- seaborn
- spacy
- transformers
- matplotlib

3.2.3 Emotion Intensity Prediction

- pandas
- numpy
- vaderSentiment
- text2emotion
- matplotlib
- os
- sys
- torch
- time
- tqdm
- sklearn
- seaborn
- spacy

3.2.4 Emotion Cause Pair Extraction

- sys
- os
- numpy
- torch
- sklearn

3.2.5 Text Summarisation

- pandas
- numpy
- tqdm
- requests
- os
- re
- torch
- transformers

CHAPTER 4: THE DATASETS

4.1 The Need for Multiple Datasets

There exist many sources of social media data. In this project, data from Twitter and Reddit will be used to analyse those depressive social media texts. Depending on the different main tasks or subtasks, appropriate data cleaning will be carried out for each task.

There is a need for multiple datasets so that the analysis of depressive text can be more generalised. Twitter datasets are usually of shorter length because the number of characters that can be posted in a tweet is limited by the founder. In contrast to Twitter, texts extracted from Reddit are much longer because there are no restrictions on the number of characters that could be written and posted. The authors can freely express themselves on Reddit. The difference in the length of texts in each dataset leads to the need for multiple datasets.

In this project, there will initially be three datasets that will be used for our analysis. However, one of which is deemed unsuitable to use and the reason for that will be explained in chapter 6 of the report on the emotion classification task. There will be another additional dataset that will be used to aid the training process for the emotion intensity prediction tasks which will be further explained in chapter 7.

4.2 Datasets used in this project

4.2.1 Toy Dataset

The toy dataset is obtained from a public GitHub repository [36]. It consists of 10314 text entries. The dataset is annotated with zeros and ones. Zeros represent non-depressive texts whereas ones represent depressive texts. There are 8000 non-depressive texts and 2314 depressive texts. This text corpus is scraped from Twitter. Table 1 shows a sample of data from this dataset in csv format.

Text	Label
"Wow, What A Tight Fit" Lmao, Shutup.	0
@theokk don't know what you could possibly mean, dear boy......	0
@shwood loved the cameo, made me chuckle! along with your SXSW talk over the past couple of shows	0
lol. i just realized my room has a color theme. green is definitely growing in on me. go green! hahaha. i still love purple though!	0
ReCoVeRiNg FrOm ThE lOnG wEeKeNd	0
The lack of this understanding is a small but significant part of what causes anxiety & depression to both feel so incredibly lonely. It's soooo easy to compare. It's so easy to invalidate ourselves because of that.	1
i just told my parents about my depression and it's so hard to get gen x people to understand that this is not something that i can control all the time or just cure with a walk or by keeping my mind busy	1
depression is something i don't speak about even going through it because it's also such a double edged sword. i love every race. even if white people had done so much i can't hate them all. my grandma is legit white! how tf can i hate, i do dislike people though.	1
Made myself a tortilla filled with pb&j. My depression is cured. Olivia:1 depression:0	1
@WorldofOutlaws I am gonna need depression meds soon, these rainouts are spinning my equilibrium out <Emoji: Pouting face> Mother Nature is being a mean B-T-H <Emoji: Face with symbols over mouth>	1

Table 1: Sample data of the Toy Dataset

4.2.2 Twitter Dataset (Short Dataset)

Similar to the Toy Dataset, the text corpus is scraped from Twitter and the dataset can be found in another public GitHub repository [37]. The annotations for the data are the same as the Toy Dataset. It consists of 3201 text entries. There are 2357 non-depressive texts and 844 depressive texts. Table 2 shows a sample of data from this dataset in csv format.

Text	Label
I really feel lk crying right now	1
I've finished a whole jar of nutella in the past hour	0
My #dark heart is filled with glee whenever I hear #thunder.	0
@dp_srk_rk It's going 3 only. He has berdych tomorrow. Nothing to worry. Easiest draw	0
Why do i always get treated like this...ugh!!!	1
It's really so sad, these stories about shootings. Makes me feel so trying to keep my head up and not crash down again. Z	1
Shakespeare Dictionary: The word 'knotty-pated' means 'block-headed, dull-witted'	0
I can't even think about football let alone watch it this afternoon	1
Watching Air Force One, good film but not paying much interest in it. Feel too rough	1
Be happy. Be confident. Be kind.\n\n #KissablesLoveSMShopmag\nAllOutDenimFor KISSMARC	0

Table 2: Sample data of the Twitter Dataset

4.2.3 Reddit Dataset (Long Dataset)

The text corpus is scraped from Reddit and the dataset can be found in a public GitHub repository [38]. Unlike the Toy and Twitter dataset, the Reddit data entries are much longer with more descriptions in it. The dataset comes in separate text files. Figure 10 and 11 shows a sample of depressive and non-depressive texts respectively.

```
Sadness is the only thing getting me through my day
I'm not really sure where to talk about this because
the people in my life don't really understand. I feel
like I messed up a lot of my life and the track it
was going on. Now I'm just very disappointed and sad
with myself. I've tried pills and therapy. However,
this might sound weird, but my sadness is the only
thing pulling me through the day and making it pass
by quickly. I don't know how to explain it because
most people associate sadness with something they
instantly want gone. Anyone else feel this way?
```

Figure 10: An entry of a depressive text in the Reddit Dataset

```
 This week was somewhat of a breakthrough in
the jobsearch.  I even got an offer for a
job I had to refuse.  On Tuesday I got two
phonecalls.  One was from Turning the Page,
an organization that does professional
development for teachers and connects
communities to schools.  They were calling
for a second interview.  The other was from
Hispanic CREO, a group that advocates for
school choice reforms.  They offered me a
(well) paid internship until something else
comes up.   I had my TTP interview
yesterday, and I think it went really well.
I don't want to count my eggs before they
hatch, though.  I have no idea what my
competition looks like!
```

Figure 11: An entry of a non-depressive text in the Reddit Dataset

To make the importing of Reddit data easier, these separate text files, depressive and non-depressive, are read and combined together to form the final Reddit dataset in csv. Figure 12 shows a code snippet to declare the folder path where the text files are in so that the Jupyter Notebook is able to read the contents of the text files. In addition, those text entries with less than 50 words are being filtered out so that this text corpus is solely for long text entries. Figure 13 and 14 shows the code snippet of how the depressive and non-depressive entries are read into the Jupyter Notebook respectively and also the filtering of text entries.

```
# root path to all the data files
ROOT = './data/text_long_raw/'
```

```
# introduce file path constant
# a mixture of reddit and blog data
DATA_Y = ROOT + 'mixed_depression/'
DATA_N = ROOT + 'mixed_non_depression/'
print(DATA_Y)
print(DATA_N)
```

```
./data/text_long_raw/mixed_depression/
./data/text_long_raw/mixed_non_depression/
```

Figure 12: Code snippet to declare the folder paths that contain the text files

```
# save all the depressed text in a dataframe
depressed_list = []

for i in tqdm(os.listdir(DATA_Y)):
    with open(DATA_Y + i, encoding="utf8") as f:
        contents = f.read()
        contents = contents.replace('\n',' ')
        # only save those data with more than 50 words
        if(len(re.findall(r'\w+', contents)) >= 50):
            depressed_list.append(contents)
len(depressed_list)
```

```
100%|██████████████████████████████| 1683/1683 [00:11<00:00, 147.91it/s]

1437
```

Figure 13: Code snippet to read in the depressive text entries with more than 50 words

```
# likewise do the same for the non-depressive text
non_depressed_list = []

for i in tqdm(os.listdir(DATA_N)):
    with open(DATA_N + i, encoding="utf8") as f:
        contents = f.read()
        contents = contents.replace('\n',' ')
        # only save those data with more than 50 words
        if(len(re.findall(r'\w+', contents)) >= 50):
            non_depressed_list.append(contents)
len(non_depressed_list)
```

```
100%|██████████████████████████████████████████████████████████████████████████| 1481/1481 [00:07<00:00, 203.41it/s]
```

```
1292
```

Figure 14: Code snippet to read in the non-depressive text entries with more than 50 words

Then, both data lists will be converted to two different dataframe, depressive text dataframe and the non-depressive text dataframe. Figure 15 and 16 shows the code snippet to convert the depressive texts list and the non-depressive texts list into their text dataframe respectively.

```
# convert the depressed list into pandas dataframe
depressed_df = pd.DataFrame(depressed_list, columns=['Text'])
# append 1 to show depressive text
depressed_df['Label'] = 1
depressed_df
```

	Text	Label
0	Just another night. Another night of feeling l...	1
1	Is it possible to fake depression? I have been...	1
2	Imagine being attractive Imagine what it would...	1
3	Best moment to have anxiety It's 3:30am, I'm t...	1
4	hi, I'm a 21 year-old male from the uk, over t...	1
...	...	...

Figure 15: Code snippet to convert the depressive text list to a dataframe

```
# convert the non-depressed list into pandas dataframe
non_depressed_df = pd.DataFrame(non_depressed_list, columns=['Text'])
# append 1 to show depressive text
non_depressed_df['Label'] = 0
non_depressed_df
```

	Text	Label
0	My male cousin stopped talking to me all of su...	0
1	Drifting away from my best friend My best frie...	0
2	You love me... I have you here by my side... ...	0
3	"...Thanks to you now I know all my dreams ...	0
4	The divorce talk The wife and I have had a ver...	0
...	...	...

Figure 16: Code snippet to convert the non-depressive text list to a dataframe

Finally, the two dataframes are concatenated together and shuffled to produce the final dataframe that is to be exported as a csv file, which is the same data format for the Toy Dataset and Short Dataset. Figure 17 shows the code to concatenate the two dataframes and Figure 18 shows the shuffling before exporting it as a csv file. The exported csv file will represent the Reddit dataset and be used in the tasks which will be further elaborated in the future chapters.

```
# merge the 2 dataframes into a dataframe
main_df = pd.concat([depressed_df, non_depressed_df])
main_df
```

	Text	Label
0	Just another night. Another night of feeling l...	1
1	Is it possible to fake depression? I have been...	1
2	Imagine being attractive Imagine what it would...	1
3	Best moment to have anxiety It's 3:30am, I'm t...	1
4	hi, I'm a 21 year-old male from the uk, over t...	1
...	...	...
1287	I hate my mum I dislike my mum. In fact, I'm d...	0
1288	Me and my brother cannot stop fighting This is...	0
1289	Pissed at my wife and mother in law We are new...	0
1290	The Glory Hole, a fun group of friends on Disc...	0
1291	Is he justified for being so mad ?? Thoughts o...	0

Figure 17: Code snippet to concatenate the two dataframes

```
# shuffle the data in the dataframe
main_df = main_df.sample(frac=1).reset_index(drop=True)
main_df
```

	Text	Label
0	Thoughts on multi-family and multi-generationa...	0
1	God "God moves towards those who need him the ...	1
2	yah.sorry i didnt write anything in the past ...	0
3	I am a 40 year old mother of two; ages 14 and ...	1
4	I find myself hating birthdays and Christmas n...	1
...	...	...
2724	SUMMER SUCKS! I know, I know...*my* opinion....	0
2725	i can't sleep i've been planning for my suicid...	1
2726	How can I help my husband? My husband is an en...	0
2727	Dude I do the right thing to have my sister on...	0
2728	Why does it feel like I want to be depressed? ...	1

Figure 18: Code snippet to shuffle the entries in the dataframe

After some processing of texts, the final Reddit Dataset consists of 2729 text entries. There are 1292 non-depressive texts and 1437 depressive texts.

4.2.4 WASSA 2017 EmoInt Dataset

The WASSA 2017 EmoInt Datasets can be obtained from the official competition website [39]. The official aim is to use the train data to build a model, then predict the emotion intensities of the corresponding test sets for the different emotions.

In this project's context, these datasets will be used as the training data for the Emotion Intensity Prediction tasks. There are originally four different emotions: anger, fear, joy and sadness, but only the anger, fear and sadness datasets will be used in this project. These text datasets came in the form of text files. Hence, to standardise the data format, the use of Jupyter Notebook and some code is needed to transform the data into the csv format. Figure 19, 20, 21 shows the first five samples of the final dataframes that are ready to be exported into csv format for the three different emotions: anger, fear and sadness respectively.

```
# rename the column header for standardization
anger_dev = anger_dev.rename(columns={1: 'Text', 2: 'Label
', 3:'Score'})
anger_dev.head()
```

	Text	Label	Score
0	@ZubairSabirPTI pls dont insult the word 'Molna'	anger	0.479
1	@ArcticFantasy I would have almost took offens...	anger	0.458
2	@IllinoisLoyalty that Rutgers game was an abom...	anger	0.562
3	@CozanGaming that's what lisa asked before she...	anger	0.500
4	Sometimes I get mad over something so minuscul...	anger	0.708

Figure 19: First five samples of the final dataframe for the emotion: anger

```
# rename the column header for standardization
fear_dev = fear_dev.rename(columns={1: 'Text', 2: 'Label
', 3:'Score'})
fear_dev.head()
```

	Text	Label	Score
0	I know this is going to be one of those nights...	fear	0.771
1	This is #horrible: Lewis Dunk has begun networ...	fear	0.479
2	@JeffersonLake speaking of ex cobblers, saw Ri...	fear	0.417
3	@1johndes ball watching & Rojo'd header wa...	fear	0.475
4	Really.....#Jumanji 2....w/ The Rock, Jack Bla...	fear	0.542

Figure 20: First five samples of the final dataframe for the emotion: fear

```
# rename the column header for standardization
sadness_dev = sadness_dev.rename(columns={1: 'Text', 2: 'Label',
3:'Score'})
sadness_dev.head()
```

	Text	Label	Score
0	@1johndes ball watching & Rojo'd header wa...	sadness	0.583
1	A pessimist is someone who, when opportunity k...	sadness	0.188
2	A .500 season is all I'm looking for at this p...	sadness	0.688
3	Stars, when you shine,\nYou know how I feel.\n...	sadness	0.292
4	All I want to do is watch some netflix but I a...	sadness	0.667

Figure 21: First five samples of the final dataframe for the emotion: sadness

4.3 Data Cleaning for each of the Tasks

With all the data being processed and exported into the standardised csv format, this subsection will take a step further to describe the data cleaning process for all the main tasks and subtasks because different tasks will require a different level of text cleaning for the data.

4.3.1 Types of Data Cleaning and the Helper Functions

There are several data cleaning strategies in this project written in helper functions, these helper functions are:

- contraction_removal
  - Replace all contractions with proper English words.
- tweet_processor
  - Remove the Uniform Resource Locator **(**URL)
  - Remove '@' and '#'
  - Remove numbers
  - Remove emojis
- clean_text
  - Lowercase all letters
  - Remove numbers
  - Remove punctuations
  - Remove extra spaces
- keep_alphabet_only
  - Only keep the alphabets
- keep_selected
  - Keep alphabets
  - Keep some basic punctuation (, . ! ?)
  - Keep numbers
- elimate_multi_letters
  - Remove letters who appeared more than twice in the text
- stopword_removal
  - Remove stopwords

- lemmatize_text
    - Lemmatise the text

4.3.2 Classification

For classification, all the above stated helper functions apply. This is because in classification, the main point of interest is the text itself, hence a ‘deep’ data cleaning is needed so that the performance of the classification model will be good. Figure 22 shows the code snippet of the helper function to clean the data for the classification task.

```python
# for the deep clean data
def deep_clean(text):
    text = contraction_removal(text)
    text = tweet_preprocessor(text, 'deep_clean')
    text = stopword_removal(text)
    text = clean_text(text,
                      removeLower=True,
                      removeNumbers=True,
                      removePunct=True,
                      removeExtraSpace=True)
    text = keep_alphabet_only(text)
    text = eliminate_multi_letters(text)
    text = lemmatize_text(text)
    return text
```

Figure 22: Code snippet to clean the data for the classification task

4.3.3 Emotion Intensity Prediction

For emotion intensity prediction, as rule-based methods such as text2emotion and VADER will be implemented, the cases and punctuations of the text matters as it accounts for the difference in the intensity scores. Figure 23 shows the code snippet of the helper function to clean the data for the emotion intensity prediction.

```python
# for the vader and t2e data
def vader_and_t2e_clean(text):
    text = contraction_removal(text)
    text = tweet_preprocessor(text, 'vader')
    text = clean_text(text,
                      removeLower=False,
                      removeNumbers=True,
                      removePunct=False,
                      removeExtraSpace=True)
    text = keep_selected(text)
    text = eliminate_multi_letters(text)
    text = lemmatize_text(text)
    return text
```

Figure 23: Code snippet to clean the data for the emotion intensity prediction task

4.3.4 Emotion-cause Pair Extraction

For emotion-cause pair extraction, the texts are not lemmatised as this will affect the coherence of the sentence. Further pre-processing of text, such as the data format will be discussed in a later chapter on emotion-cause pair extraction. Figure 24 shows the code snippet of the helper function to clean the data for emotion-cause pair extraction.

```python
def ecpe_clean(text):
    text = contraction_removal(text)
    text = tweet_preprocessor(text, 'ecpe')
    text = clean_text(text,
                      removeLower=False,
                      removeNumbers=False,
                      removePunct=False,
                      removeExtraSpace=True)
    text = eliminate_multi_letters(text)
    return text
```

Figure 24: Code snippet to clean the data for the emotion-cause pair extraction task

CHAPTER 5: API AND LIBRARIES USED IN THIS PROJECT

This chapter provides some description on some of the APIs and libraries that will be used in this project.

5.1 Sentic APIs

Sentic APIs are a suite of APIs that perform various sentiment analysis tasks created by the Sentic Team [40]. All the APIs are based on the Sentic computing framework which uses an ensemble of symbolic AI (SenticNet) and subsymbolic AI (Deep Learning) [41]. The main aim of SenticNet is to make the conceptual and affective information conveyed by natural language easily understandable by the machines [42].

In this project, the Sentic API on depression identification will be used.

Depression Identification

This API enhances text representation with lexicon-based sentiment scores and latent topics and uses relation networks to identify depression with a related risk indicator. The API takes in a piece of text and outputs a depression score between zero to 100, where zero implies no depression detected and 100 implies high depression detected [40]. Figure 25 shows the high-level diagram of the architecture. This API will be used in Chapter 7 of the report on Emotion Intensity Prediction.

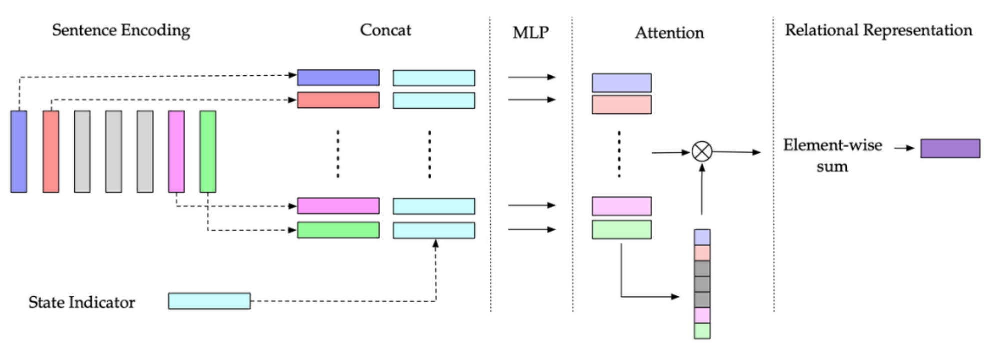

Figure 25: High level diagram of depression identification Sentic API

5.2 DeepAI Text Summarisation API

This API cuts down the size of a document by retaining the most relevant sentences. It aims to reduce the size of each entry to 20% of the original [43]. This API will be used in Chapter 8 on Text Summarisation.

5.3 Valence Aware Dictionary and Sentiment Reasoner (VADER)

VADER is a sentiment lexicon and rule-based sentiment analysis tool that is specially attuned to sentiments expressed in social media. A sentiment lexicon is a list of lexical features, such as words, which are usually labelled according to their semantic orientation as either positive or negative. In addition, VADER also reflects how positive or negative a sentiment is [44]. Figure 26 illustrates an example of using the library. This python library will be used in Chapter 7 on Emotion Intensity Prediction.

```
[ ]  # import vader sentiment intensity analyser
     from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer

     # testing VADER
     analyser = SentimentIntensityAnalyzer()
     score = analyser.polarity_scores("I feel so SAD")
     print(score)
     print(score['neg'])
```

```
{'neg': 0.621, 'neu': 0.379, 'pos': 0.0, 'compound': -0.7102}
0.621
```

Figure 26: An example of using the VADER library

5.4 Text2Emotion

Text2Emotion is the python package to look for the appropriate emotions embedded in the text data [45]. It has 5 basic emotions, they are: angry, fear, happy, sad and surprise. Figure 27 illustrates an example of using the library. This python library will be used in Chapter 7 on Emotion Intensity Prediction.

```python
# import text2emotion library
import text2emotion as te

# sample text
text = "I was asked to sign a third party contract a week out from stay. \
If it wasn't an 8 person group that took a lot of wrangling \
I would have cancelled the booking straight away. \
Bathrooms - there are no stand alone bathrooms. \
Please consider this - you have to clear out the main bedroom to \
use that bathroom. Other option is you walk through a different \
bedroom to get to its en-suite. Signs all over the apartment - \
there are signs everywhere - some helpful - some telling you rules. \
Perhaps some people like this but It negatively affected \
our enjoyment of the accommodation. Stairs - lots of them \
- some had slightly bending wood which caused a minor injury."
# sample usage of text2emotion
te.get_emotion(text)
```

```
{'Angry': 0.12, 'Fear': 0.42, 'Happy': 0.04, 'Sad': 0.33, 'Surprise': 0.08}
```

Figure 27: An example of using the text2emotion library

CHAPTER 6: MAIN TASK 1 - EMOTION CLASSIFICATION

This chapter describes the use of the datasets mentioned in chapter 4 to perform the classification task. An in-depth description of the implementation will be stated below. Both the model and data centric approach will be taken for this task.

6.1 Variations in the Experiment

In this classification task, there will be four datasets being experimented, they are:

- Toy Dataset
- Twitter Dataset (Short Data)
- Reddit Dataset (Long Data)
- Twitter Dataset + Reddit Dataset (Short + Long Data)

The models experimented are:

- Unidirectional LSTM
- Bidirectional LSTM
- Unidirectional GRU
- Bidirectional GRU
- BERT

A total of 20 models will be built to determine the best performing one for each dataset based on the various metrics such as validation accuracy, F1 score and AUC.

6.2 Hyperparameters held constant for the different datasets

For the sequence models, the hyperparameters were experimented and these are the hyperparameters that produces the best performances in general for the different sequence model architectures:

- Number of Epochs: 15
- Learning Rate: 5e-04
- Hidden Dimension: 512
- Number of RNN layers: 4
- Dropout: 0.5
- Batch Size: 32
- Optimiser: Adam

For the transformer model BERT, the hyperparameters are:

- Number of Epochs: 2
- Learning Rate: 5e-05
- Batch Size: 16
- Optimiser: AdamW

6.3 Implementation

The datasets were splitted into their corresponding train and validation sets individually. The train_test_split built in function from the sklearn library is used to split all the datasets into 80% train data and 20% validation data.

6.3.1 Data Fields

There will be three fields (or three columns) in all the four datasets, they are:

- Text
    - Raw text entries without text pre-processing
- Label
    - The label for the text entries, 0 for non-depressive content and 1 for depressive content
- text_cleaned
    - Text entries after text pre-processing

6.3.2 Data Loaders

The codes to build the sequence models and the BERT model were written in PyTorch, an open-source machine learning framework that accelerates the path from research prototyping to production deployment. As such, there is a need to write a custom dataset loader to load the data into Google Colaboratory for analysis. Figure 28 and Figure 29 shows the data loader implementation for the sequence models and the BERT to load the fields in respectively.

```
[ ] class CustomDataset(data.Dataset):
        def __init__(self, df, fields, is_test=False, **kwargs):
            examples = []
            for i, row in df.iterrows():
                label = row.Label #if not is_test else None
                text = row.text_cleaned
                examples.append(data.Example.fromlist([text, label], fields))

            super().__init__(examples, fields, **kwargs)

        @staticmethod
        def sort_key(ex):
            return len(ex.text)

        @classmethod
        def splits(cls, fields, train_df, val_df=None, test_df=None, **kwargs):
            train_data, val_data, test_data = (None, None, None)
            data_field = fields

            if train_df is not None:
                train_data = cls(train_df.copy(), data_field, **kwargs)
            if val_df is not None:
                val_data = cls(val_df.copy(), data_field, **kwargs)
            if test_df is not None:
                test_data = cls(test_df.copy(), data_field, False, **kwargs)

            return tuple(d for d in (train_data, val_data, test_data) if d is not None)
```

Figure 28: Data Loader for the sequence models

```
def create_data_loader(root,train_filename,dev_filename,batch_size=16):
    X_train, y_train, X_val, y_val, X_ = data_handler(root, train_filename, dev_filename)

    # Concatenate train data and test data
    all_texts = np.concatenate([X_train, X_val])

    # Encode our concatenated data
    encoded_texts = [tokenizer.encode(sent, add_special_tokens=True) for sent in all_texts]

    # Find the maximum length
    max_len = max([len(sent) for sent in encoded_texts])
    #print('Max length: ', max_len)

    # Print sentence 0 and its encoded token ids
    token_ids = list(preprocessing_for_bert([X_[0]])[0].squeeze().numpy())
    #print('Original: ', X_[0])
    #print('Token IDs: ', token_ids)

    # Run function `preprocessing_for_bert` on the train set and the validation set
    print('Tokenizing data...')
    train_inputs, train_masks = preprocessing_for_bert(X_train)
    val_inputs, val_masks = preprocessing_for_bert(X_val)

    # Convert other data types to torch.Tensor
    train_labels = torch.tensor(y_train)
    val_labels = torch.tensor(y_val)

    # Create the DataLoader for our training set
    train_data = TensorDataset(train_inputs, train_masks, train_labels)
    train_sampler = RandomSampler(train_data)
    train_dataloader = DataLoader(train_data, sampler=train_sampler, batch_size=batch_size)

    # Create the DataLoader for our validation set
    val_data = TensorDataset(val_inputs, val_masks, val_labels)
    val_sampler = SequentialSampler(val_data)
    val_dataloader = DataLoader(val_data, sampler=val_sampler, batch_size=batch_size)

    return train_dataloader, val_dataloader, y_val
```

Figure 29: Data Loader for the BERT model

6.3.3 Model Building

To build the models, the implementation of python classes is taken. For the sequence model, the model variant is being passed as parameters of the class implementation for ease of calling it for the various sequence models. Some variants are: the architecture of the model (LSTM vs GRU) and bidirectional or unidirectional. Figure 30 and 31 illustrates the code snippets of the model definitions for the sequence models and BERT model respectively.

```
class RNN(nn.Module):
    def __init__(self, vocab_size, embedding_dim, hidden_dim, output_dim, n_layers,
                 bidirectional, rnn_type, dropout, pad_idx):
        super().__init__()

        self.embedding = nn.Embedding(vocab_size, embedding_dim, padding_idx = pad_idx)
        self.rnn_type = rnn_type

        # LSTM
        if self.rnn_type == 'lstm':
            self.rnn = nn.LSTM(embedding_dim,
                               hidden_dim,
                               num_layers=n_layers,
                               bidirectional=bidirectional,
                               dropout=dropout,
                               batch_first=True)
        # GRU
        else:
            self.rnn = nn.GRU(embedding_dim,
                               hidden_dim,
                               num_layers=n_layers,
                               bidirectional=bidirectional,
                               dropout=dropout,
                               batch_first=True)

        self.fc1 = nn.Linear(hidden_dim * 2, hidden_dim)
        self.fc2 = nn.Linear(hidden_dim, 1)

        self.dropout = nn.Dropout(dropout)

    def forward(self, text, text_lengths):
        embedded = self.embedding(text)

        # pack sequence
        packed_embedded = nn.utils.rnn.pack_padded_sequence(embedded,
                                                            text_lengths.to('cpu'),
                                                            batch_first=True)
        if self.rnn_type == 'lstm':
            packed_output, (hidden, cell) = self.rnn(packed_embedded)
        else:
            packed_output, hidden = self.rnn(packed_embedded)

        hidden = self.dropout(torch.cat((hidden[-2,:,:], hidden[-1,:,:]), dim = 1))

        output = self.fc1(hidden)
        output = self.dropout(self.fc2(output))

        return output
```

Figure 30: Model implementation for the sequence models

```python
class BertClassifier(nn.Module):
    def __init__(self, freeze_bert=False):
        super(BertClassifier, self).__init__()
        # Specify hidden size of BERT, hidden size of our classifier, and number of labels
        D_in, H, D_out = 768, 50, 2

        # Instantiate BERT model
        self.bert = BertModel.from_pretrained('bert-base-uncased')

        # Instantiate an one-layer feed-forward classifier
        self.classifier = nn.Sequential(
            nn.Linear(D_in, H),
            nn.ReLU(),
            #nn.Dropout(0.5),
            nn.Linear(H, D_out)
        )

        # Freeze the BERT model
        if freeze_bert:
            for param in self.bert.parameters():
                param.requires_grad = False

    def forward(self, input_ids, attention_mask):
        # Feed input to BERT
        outputs = self.bert(input_ids=input_ids,
                            attention_mask=attention_mask)

        # Extract the last hidden state of the token `[CLS]` for classification task
        last_hidden_state_cls = outputs[0][:, 0, :]

        # Feed input to classifier to compute logits
        logits = self.classifier(last_hidden_state_cls)

        return logits
```

Figure 31: Model implementation for the BERT models

6.3.4 Model Performances

After loading the dataset and building the models, the models are trained with the train data and evaluated on the validation data for each of the four datasets. Table 3, 4, 5, 6 shows the model performances based on various metrics for the Toy dataset, Twitter dataset, Reddit dataset and the combined Twitter and Reddit dataset respectively. Figure 32, 33, 34, 35 shows the ROC curves for the models trained on Toy dataset, Twitter dataset, Reddit dataset and the combined Twitter and Reddit dataset for the sequence model respectively. Figure 36 shows the ROC curves for the BERT model trained on all the four datasets.

Toy Dataset

Model	Train Loss	Validation Loss	Validation Accuracy	F1 Score (Marco Average)	F1 Score (Weighted Average)	AUC
Uni LSTM	0.35462	0.06966	0.98384	0.9834	0.9834	0.99234
Bi LSTM	0.35311	0.07202	0.9806	0.9801	0.9801	0.99436
Uni GRU	0.35064	0.06779	0.98168	0.9812	0.9812	0.99267
Bi GRU	0.35798	0.08556	0.97953	0.979	0.979	0.98916
BERT	0.007685	0.008193	0.9989	0.9989	0.9989	**1**

Table 3: Model Performances on the Toy Dataset

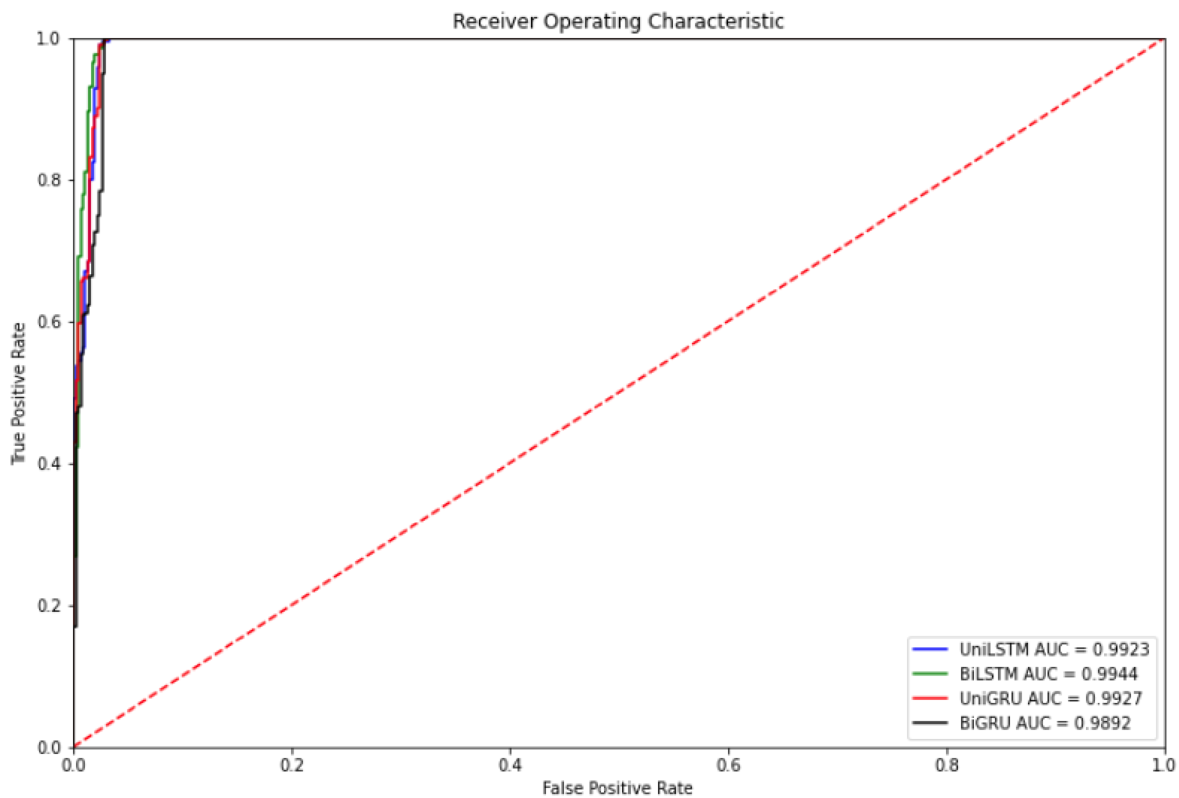

Figure 32: ROC curve for the models trained on the Toy Dataset

Twitter Dataset

Model	Train Loss	Validation Loss	Validation Accuracy	F1 Score (Marco Average)	F1 Score (Weighted Average)	AUC
Uni LSTM	0.44219	0.57769	0.72768	0.722	0.722	0.80175
Bi LSTM	0.39676	0.57645	0.73539	0.7362	0.7362	0.80849
Uni GRU	0.3725	0.67257	0.70414	0.7036	0.7036	0.78461
Bi GRU	0.48157	0.55496	0.70698	0.7063	0.7063	0.79508
BERT	0.35721	0.48968	0.7883	0.7874	0.7874	**0.8726**

Table 4: Model Performances on the Twitter Dataset

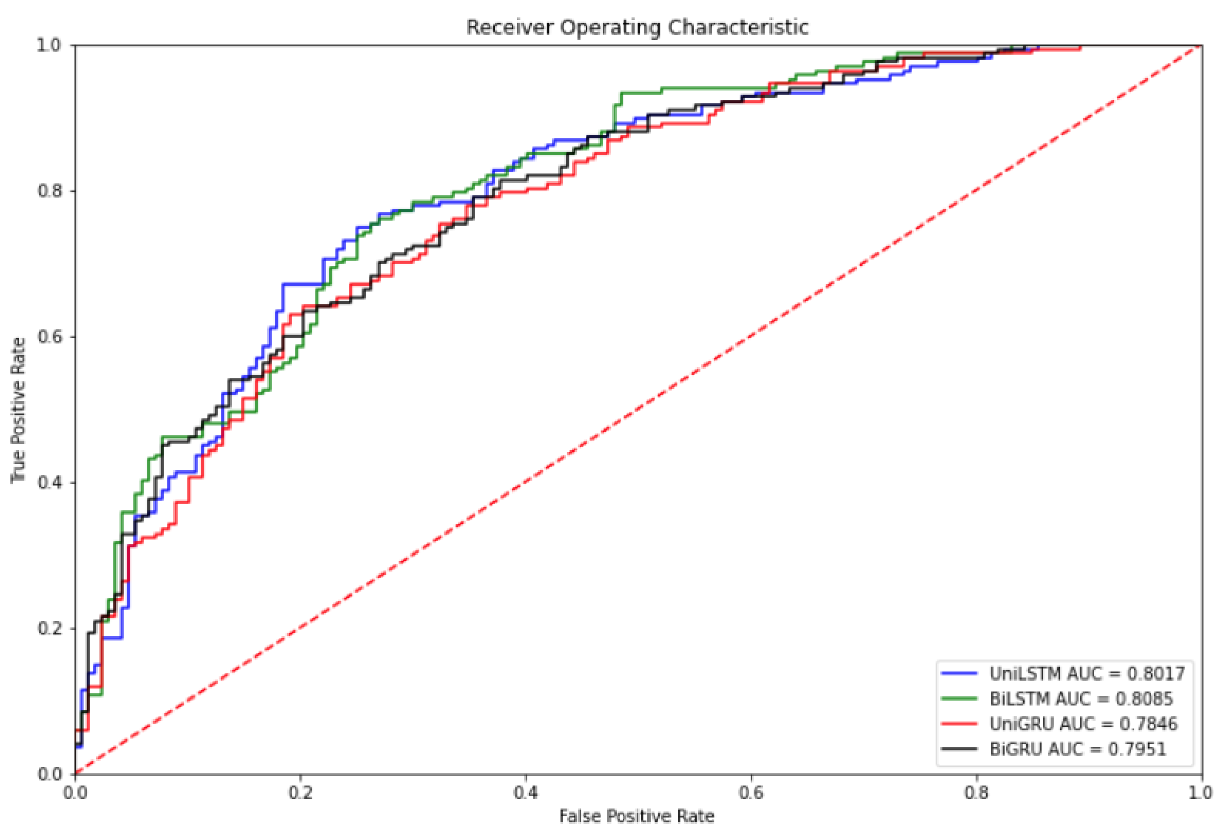

Figure 33: ROC curve for the models trained on the Twitter Dataset

Reddit Dataset

Model	Train Loss	Validation Loss	Validation Accuracy	F1 Score (Marco Average)	F1 Score (Weighted Average)	AUC
Uni LSTM	0.46182	0.40975	0.8364	0.8272	0.8272	0.89714
Bi LSTM	0.36291	0.4242	0.86642	0.8776	0.8776	0.9408
Uni GRU	0.4387	0.40307	0.81985	0.8072	0.8071	0.90907
Bi GRU	0.36831	0.35474	0.85723	0.8679	0.8679	0.93939
BERT	0.2049	0.37976	0.8832	0.897	0.897	**0.95**

Table 5: Model Performances on the Reddit Dataset

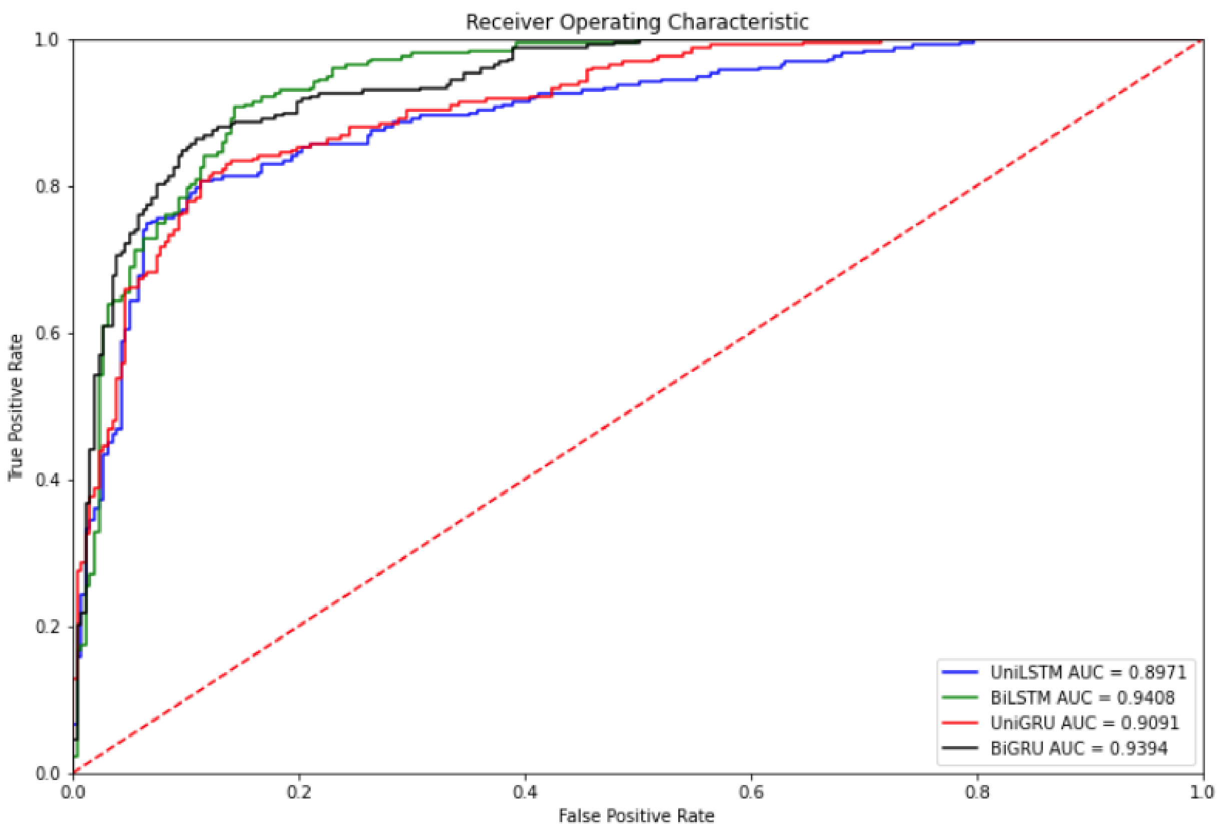

Figure 34: ROC curve for the models trained on the Reddit Dataset

Combined Twitter and Reddit Dataset

Model	Train Loss	Validation Loss	Validation Accuracy	F1 Score (Marco Average)	F1 Score (Weighted Average)	AUC
Uni LSTM	0.41167	0.48861	0.76532	0.7632	0.7632	0.85459
Bi LSTM	0.43118	0.46275	0.80467	0.8033	0.8033	0.86632
Uni GRU	0.4517	0.5068	0.7517	0.7515	0.7515	0.83117
Bi GRU	0.37036	0.47504	0.78499	0.7832	0.7832	0.877
BERT	0.26311	0.44791	0.8241	0.8385	0.8385	**0.9061**

Table 6: Model Performances on the Combined Twitter and Reddit Dataset

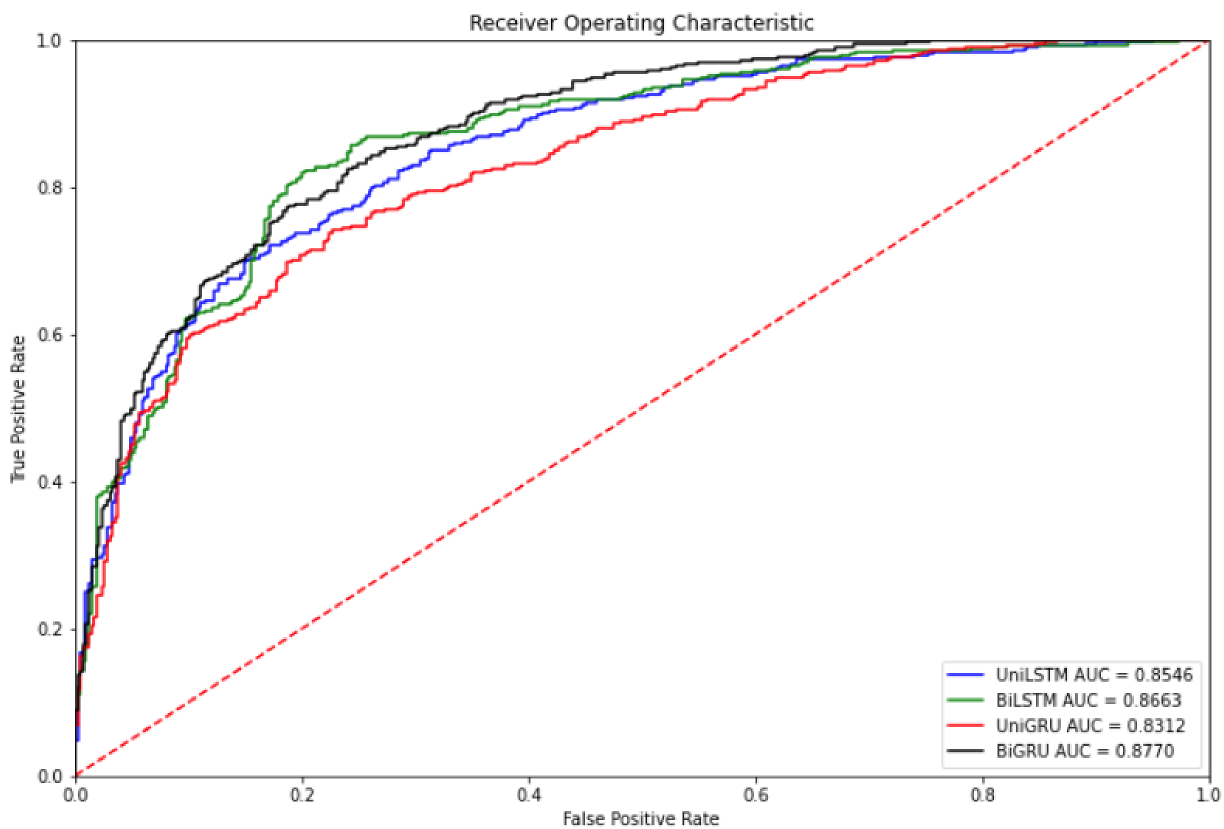

Figure 35: ROC curve for the models trained on the combined Twitter and Reddit Dataset

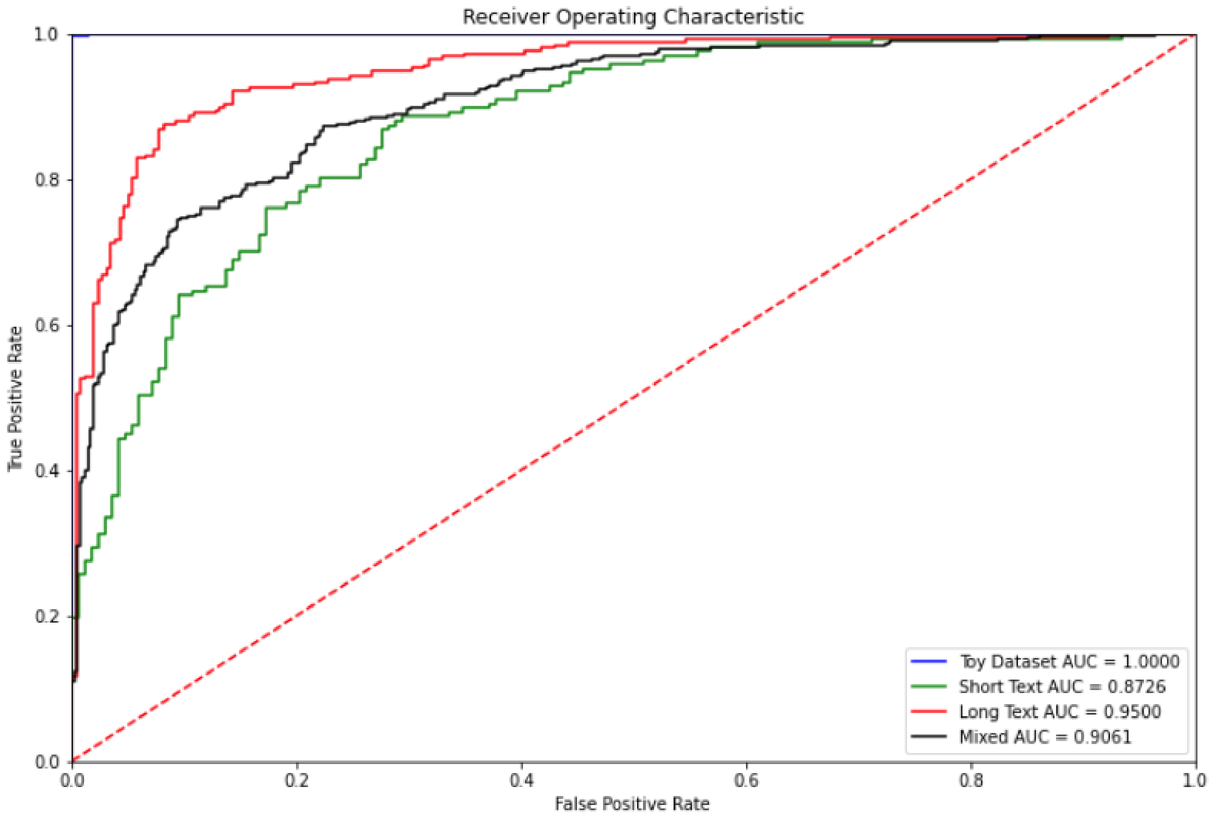

Figure 36: ROC curves for the BERT model trained on all the four datasets

From the above tables and ROC curves, the BERT transformer model outperformed all the sequence models in the validation loss, validation accuracy, F1 score and AUC for all four datasets. This is due to the fact that sequence models are trained from the leftmost word to the rightmost word to predict the next word. If bidirectional, the sequence models will train from both left-to-right to predict the next word, and also to train from right-to-left to predict the previous word. But none of them took a look at both ways at the same time. In contrast, the BERT model is made to learn from words in all positions and as such, looking at the entire sentences at one go, making the model more accurate in its predictions and hence with a better performance as compared to sequence models [46].

To have a closer analysis at the performance of the classification models, confusion matrices can be plotted for all the four datasets that are trained on the BERT model. Table 7, 8, 9, and 10 shows the confusion matrix for the BERT model trained on the toy dataset, Twitter dataset, Reddit dataset and the combined Twitter and Reddit dataset respectively.

Toy Dataset - BERT

Actual \ Predicted	Non-depressive (‘0’)	Depressive (‘1’)
Non-depressive (‘0’)	453	0
Depressive (‘1’)	1	451

Table 7: Confusion Matrix for the toy dataset trained on the BERT model

Twitter Dataset - BERT

Actual \ Predicted	Non-depressive (‘0’)	Depressive (‘1’)
Non-depressive (‘0’)	132	35
Depressive (‘1’)	36	131

Table 8: Confusion Matrix for the Twitter dataset trained on the BERT model

Reddit Dataset - BERT

Actual \ Predicted	Non-depressive (‘0’)	Depressive (‘1’)
Non-depressive (‘0’)	237	21
Depressive (‘1’)	32	225

Table 9: Confusion Matrix for the Reddit dataset trained on the BERT model

Combined Twitter and Reddit Dataset - BERT

Actual \ Predicted	Non-depressive (‘0’)	Depressive (‘1’)
Non-depressive (‘0’)	345	80
Depressive (‘1’)	57	367

Table 10: Confusion Matrix for the combined Twitter and Reddit dataset trained on the BERT model

Generally, from the above tables, most of the predictions are correct with approximately 10-20% misclassification for each of the classes in the Twitter, Reddit and the combined dataset. However, there is only one misclassification in the Toy dataset, which contributes to only 0.110% error (1/(453+451+1)). This scenario is too good to be true. As such, there is a need to re-examine the suitability of using the toy dataset for future analysis.

6.4 Re-examining the Toy Dataset

There seemed to be a flaw in this dataset due to the 'near to perfect' validation accuracy, F1 score and also the AUC on the toy dataset as seen in Table 3. As such, two methods will be used to determine if the toy dataset is reliable and can it still be used for the other tasks ahead. These two methods are:

- Rough Observation
  - Get a sample of the data and find the common trend in the data while looking at the label at the same time.
- Inter-annotator agreement
  - Get two annotators to annotate the whole dataset and see how much agreement they have with each other and also how much agreement they have with the ground truth labels by measuring the Cohen-Kappa Score. The higher the score, the more agreement the annotators have or the more agreement there is between an annotator and the ground truth label.

6.4.1 Rough Observation

Table 11 and Table 12 shows a sample of the toy dataset having the class depressive ('1') and non-depressive ('0') respectively.

Text	Label
The Oil that has the Potential to Fight Migraines, Depression, Anxiety, & Even Cancer http://www.healthy-holistic-living.com/oil-potential-cure-migraines-depression-anxiety-even-cancer.htmlÃ‚Â Ã¢Â€Â¦	1
@ the ppl on my TL that liked the tweet about how self-care will cure depression: which essential oils will stop my hallucinations & paranoid delusions?	1
I call BS if anyone deserves credit its president Obama for putting the brakes on the worse recession in our history that almost lead to a depression if not for Obama & his adms getting the economy that GWB & the reps had managed to almost demolish back on its tracks. #REALNEWS	1
my depression: https://twitter.com/kanyewest/status/989142253468708864Ã‚Â Ã¢Â€Â¦	1
Social support, rest, ritual, food, storytelling, and touch are all common among cultural practices for #postpartum #depression. F Parks #GOLDQuotes #PPD #PPMAD #maternalhealth	1
A fat ass won't help u keep a man. Get ahead in ur career, unless ur a stripper. It won't help ur depression, save ur friendships. Chile.	1
The Home Office rejected and rejected and rejected...I entered a phase of total depression...We told them all this in 2014! Why didn't they listen? Sam, 41 years in the UK from Sierra Leone #HostileEnvironment #LegalAid #IndependentReview #migrantvoices https://www.theguardian.com/commentisfree/2018/apr/25/windrush-scandal-immigration-legal-aidÃ‚Â Ã¢Â€Â¦	1
fren: i have depressionme: have u been diagnosed?fren: i know i have depressionme: yea but have u been diagnosed??fren: no but i know i dome: how do u knowfren: bECAUSE *tears up* IM JUST SO SAD ALL THE TIME AND MY LIFE SUCKS *fake weeping sounds*	1
Bring on the post festival depression <Emoji: Sign of the horns (light skin tone)>	1
Any more 'winning' and we'll have an economic depression by noon. https://twitter.com/Vox_Democracy/status/989165030263410689Ã‚Â Ã¢Â€Â¦	1

Table 11: Text with depressive label

Text	Label
Good morning everyone	0
Busy rest of the day...meeting with prospective clients, college students who want to be "Collegepreneurs" . Later!	0
@kathrynryn not since friday but its all good its all good	0
@alecstanworth That's nothing: http://bit.ly/better-bragging-rights and she resigned too	0
slept too much, so, no school to me bitchhhh	0
@insanityreport hmmm...so, what industry in? maybe you should be my next supervisor I've probably had all women for a reason ...	0
@jamieyork Pretty good games, when you can get them running over 20 FPS. Preferred the first one myself. Worth picking up.	0
#musicmonday: Playlist: 80s Music. Now playing: Metallica - Fade to Black	0
@DJNEPTUNE I know what you mean... XOXO	0
@BBBaumgartner You can do it!!! Have a fun time golfing	0

Table 12: Text with non-depressive label

From Table 11 and Table 12, as long as the word ‘depression’ appears in the text, that entry will be labelled as ‘depressive’ text despite not having the emotion of ‘depression’ in the text, such as the second and last entry of Table 11. In contrast, looking at the second entry of Table 12, that piece of text does contain some emotion of ‘depression’ because of the phrase ‘very very sad’, but it is labelled as ‘non-depressive’ as the word ‘depression’ is not used. This may cause the model to be inaccurate in its prediction on a real piece of depressive text, especially one that does not use the word ‘depression’ at all but actually contains the emotion ‘depression’.

6.4.2 Inter-annotator Agreement

Another method is to do inter-annotator agreement between the two annotators and the ground truth labels. The following pairs of annotators will be:

- Annotator 1 vs Annotator 2
- Annotator 1 vs Ground Truth
- Annotator 2 vs Ground Truth

The two annotators are undergraduates from the School of Computer Science and Engineering (SCSE). The implementation of a helper function is shown in Figure 37 to compute the Cohen's Kappa coefficient and the confusion matrix of the labels for different pairs of annotations.

```
def get_score(annotator_a, annotator_b, row_name, col_name):
    # cohen-kappa score
    print(cohen_kappa_score(annotator_a, annotator_b))

    # seaborn confusion matrix
    cm_sns = pd.crosstab(annotator_a, annotator_b, rownames=[row_name], colnames=[col_name])
    sns.heatmap(cm_sns, annot=True, fmt="d")
```

Figure 37: Helper function to show the Cohen's Kappa coefficient and also the confusion matrix of a pair of annotation

Toy Dataset

Table 13 below shows the annotator pair and also the Cohen's Kappa Coefficient.

Annotator A	Annotator B	Cohen's Kappa Coefficient
Annotator 1	**Annotator 2**	**0.81073**
Ground Truth	**Annotator 1**	**0.20354**
Ground Truth	**Annotator 2**	**0.19027**

Table 13: Annotator pair and the corresponding Cohen's Kappa Coefficient for the Toy Dataset

From table 13, there seemed to be quite a good agreement between the two annotators in terms of classifying if a text is depressive or not based on the text data, with a score larger than 80%. However, there seemed to be a huge disagreement between both the annotator and the ground truth label of a Cohen's Kappa Coefficient of barely just 20%. This implies that the ground truth labels are inaccurate for the text data given. As such, the hypothesis made in the rough observation might be correct, where the texts are labelled depressive if the word 'depression' appears in the text data and non-depressive if it does not appear.

In conclusion, the toy dataset is omitted as it is deemed unsuitable for this task and also the subsequent tasks discussed in the future chapters.

Other datasets

Table 14 and 15 below shows the annotator pair and also the Cohen's Kappa Coefficient for the Twitter dataset and the Reddit dataset respectively.

Twitter Dataset

Annotator A	Annotator B	Cohen's Kappa Coefficient
Annotator 1	Annotator 2	0.81157
Ground Truth	Annotator 1	0.80240
Ground Truth	Annotator 2	0.80838

Table 14: Annotator pair and the corresponding Cohen's Kappa Coefficient for the Twitter Dataset

Reddit Dataset

Annotator A	Annotator B	Cohen's Kappa Coefficient
Annotator 1	Annotator 2	0.83622
Ground Truth	Annotator 1	0.80934
Ground Truth	Annotator 2	0.80156

Table 15: Annotator pair and the corresponding Cohen's Kappa Coefficient for the Reddit Dataset

From Table 14 and 15, all the annotator pairs have Cohen's Kappa Coefficient score of above 80%. This implies that there is a good agreement between the annotators, as well as good agreement between the individual annotator and the ground truth for both the Twitter and Reddit dataset. In conclusion, the Twitter and Reddit datasets are deemed suitable to use for this task and the other tasks which will be explained in future chapters.

6.5 Conclusion for this chapter on Emotion Classification

In this chapter, both the datasets and the models are analysed for the Emotion Classification task. In general, the BERT model performs the best for the datasets used for emotion classification. Also, the toy dataset label is not reflective of the true emotion of 'depression' in the text entries based on rough observation and inter-annotator agreement. Hence, the toy dataset will not be used for future tasks. On the other hand, the other datasets such as the Twitter and Reddit dataset are deemed suitable for this task as well as the other tasks in future chapters as the Cohen's Kappa Coefficient score for the annotation pair between the annotators and the pair between individual annotator and the ground truth exceeds 80%.

In the next chapter on the Emotion Intensity Prediction task, the primary datasets used for analysis will only be the Twitter dataset and the Reddit dataset.

CHAPTER 7: MAIN TASK 2 - EMOTION INTENSITY PREDICTION

The previous chapter describes emotion classification, whether a particular text entry is depressive or not. However, emotion classification cannot describe the intensity of depression for a particular text. Detecting how depressed one is from text is crucial as different intensity of depression requires different treatment. This chapter introduces the concept of emotion intensity prediction, where the input is a piece of text and the output is a depression score. In this chapter, only those data that are labelled as depressive will be considered.

With the lack of intensity scores in both the Twitter and Reddit datasets, there is no way to perform a regression from the input data to the output, since there is no output data of intensity scores. Fortunately, there are two other methods to obtain the depression intensity scores for the text entries even without the intensity scores.

The two methods are:

1. Combination of symbolic and sub-symbolic approach
2. Using Sentic API on Depression Identification

7.1 Method 1: Combination of symbolic and sub-symbolic approach

In this method, there will be three components to make up a composite depression score for the text entries.

7.1.1 Weightage Distribution of the Composite Score

The three components and their corresponding weightages are:

1. VADER (negative) - 20% (Symbolic)
2. Text2Emotion - 40% (Symbolic)
3. Score Prediction using the WASSA 2017 EmoInt dataset - 40% (Sub-symbolic)

Breaking down the weightages further to its individual components, the weightages will be:

1. VADER (negative) - 20%
2. Text2Emotion
    a. Angry - 5%
    b. Surprise - 5%
    c. Fear - 5%
    d. Sadness - 25%
    e. Happy - 0%
3. Score Prediction using the WASSA 2017 EmoInt dataset
    a. Dataset on anger score - 5%
    b. Dataset on fear score - 5%
    c. Dataset on sadness score - 30%
    d. Dataset on happy score - Not used (0%)

7.1.2 Rationale for this Weight Distribution

As depression is considered a negative emotion, VADER makes it suitable for one of the components in the depression metric because it has a negative score and it is a rule-based method that takes into account of the punctuations of the sentences as well as whether the words are in uppercase or lowercase to produce the negativity score of the text [44].

Text2Emotion python library is also taken into consideration because it is a rule-based method that can produce scores for the five basic emotions: angry, happy, sad, fear, surprise, making the composition of the depression scores more robust [45].

Using the WASSA 2017 EmoInt dataset as the train set for our test sets (Twitter and Reddit data) because the WASSA 2017 EmoInt dataset has labelled scores for the respective emotions. Although there is no direct depression score from this dataset, the combination of scores from anger, fear and sadness can be used to estimate the depression scores for the text entries in the Twitter and Reddit data [39].

The given weightages for each of the emotions in Text2emotion [45] and the WASSA 2017 EmoInt dataset [39] are inspired by the Sentic Hourglass Model [41] [42] and also the Plutchik's Wheel of Emotion [47]. Figure 38 and Figure 39 shows the Sentic Hourglass Model and the Plutchik's Wheel of Emotion respectively.

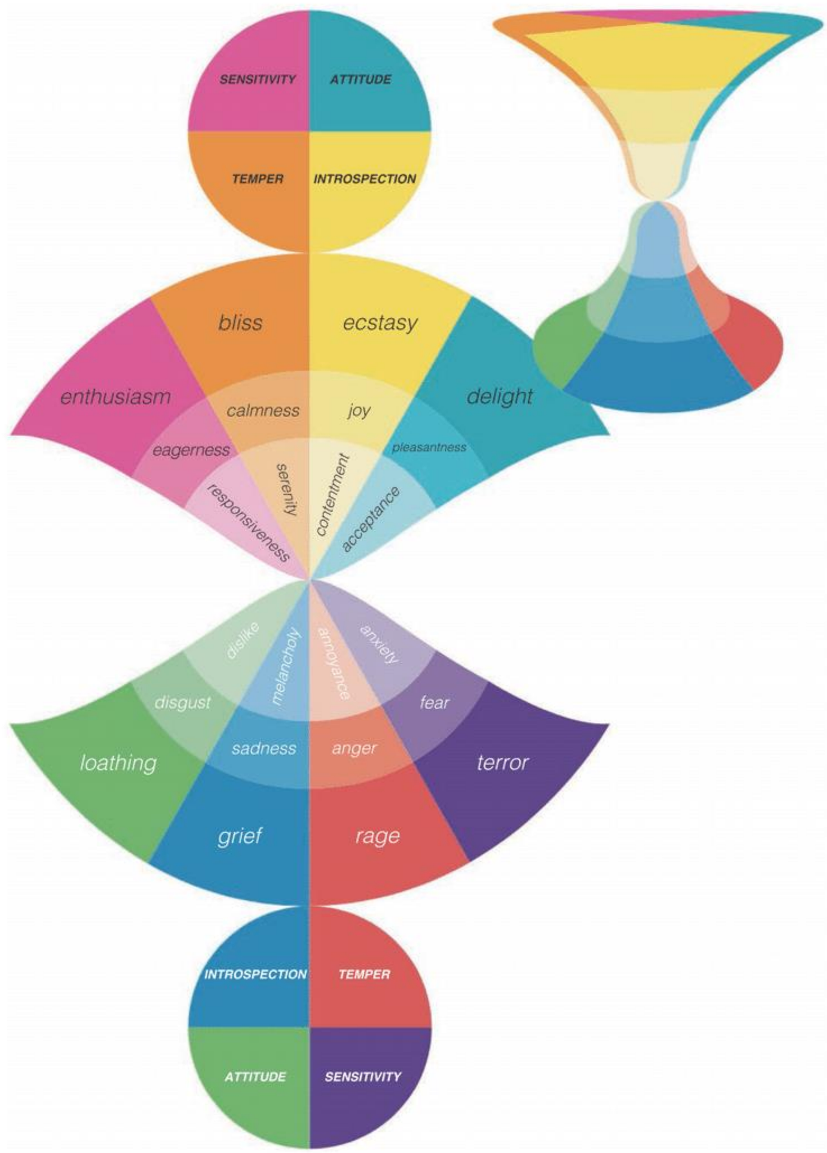

Figure 38: Sentic Hourglass Model

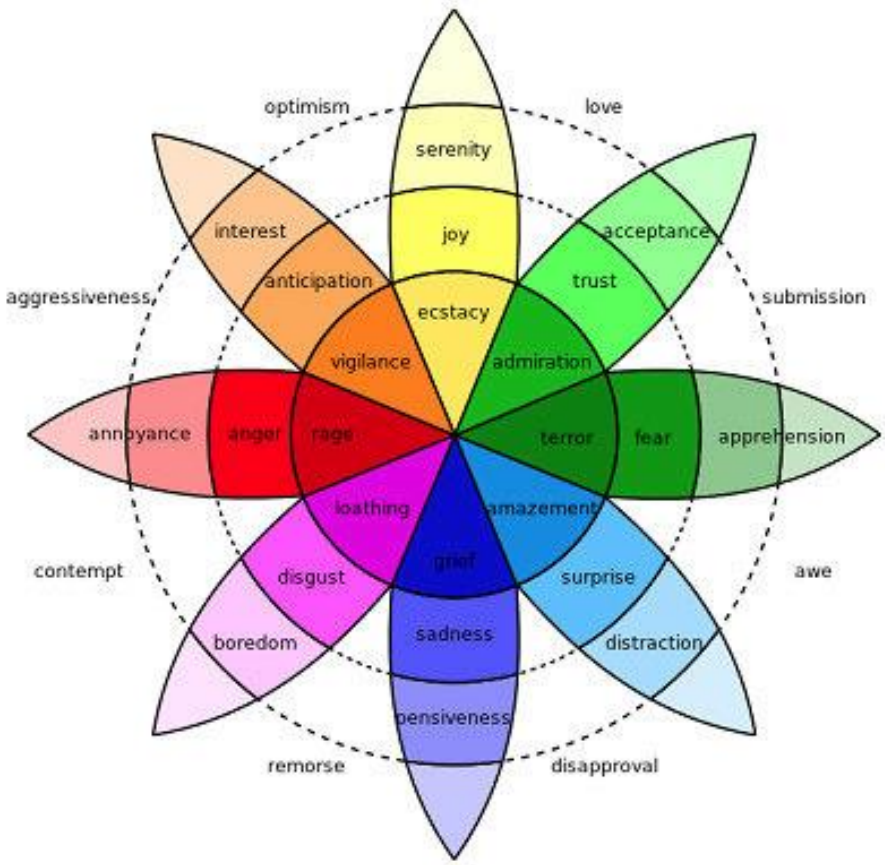

Figure 39: Plutchik's Wheel of Emotion

As depression is defined as a constant feeling of sadness and loss of interest, and may also contain a mixture of other feelings [48], the weightage of sadness is much higher than the other emotions. Also, as seen from Figure 38 and Figure 39, the other emotions such as 'anger', 'fear' and 'surprise' are near to sadness, hence these emotions took a small amount of weightage in the final score. In contrast, the emotion 'happy' is at the direct opposite side of the emotion 'sad' in both diagrams, therefore, no weightage is allocated to the emotion 'happy' in text2emotion and the WASSA 2017 EmoInt data on happiness score.

7.1.3 Implementation

VADER

For the VADER component, only the negative polarity score will be taken because depression usually bears a negative polarity in texts. Figure 40 and 41 shows the code snippet of how the sentences are being passed into the VADER library functions and how the values are being appended to the dataframe for the Twitter and Reddit dataset respectively.

```
# create a list to store all negative values from VADER for the short text
VADER_score_list = []

# do the VADER analysis
for sentence in short_data.text_cleaned_t2e_vader.tolist():
    score = analyser.polarity_scores(sentence)
    VADER_score_list.append(score['neg'])
```

```
# create dataframe to store short text and the vader scoring
rule_base_scoring_short_df = pd.DataFrame()
rule_base_scoring_short_df['text_cleaned'] = short_data.text_cleaned_t2e_vader
rule_base_scoring_short_df['vader_neg'] = VADER_score_list
rule_base_scoring_short_df.head()
```

	text_cleaned	vader_neg
0	I get to spend New Year is home again alone an...	0.306
1	Depressed and lonely Stuck in a deep, never en...	0.527
2	Learning to pretend to have a good time had be...	0.054
3	So far he stop texting me after I said somethi...	0.103
4	sigh ?? I have not cried so much I am in so mu...	0.216

Figure 40: VADER on the Twitter dataset

```
# create a list to store all negative values from VADER for the long text
VADER_score_list = []

# do the VADER analysis
for sentence in long_data.text_cleaned_t2e_vader.tolist():
    score = analyser.polarity_scores(sentence)
    VADER_score_list.append(score['neg'])
```

```
# create dataframe to store long text and the vader scoring
rule_base_scoring_long_df = pd.DataFrame()
rule_base_scoring_long_df['text_cleaned'] = long_data.text_cleaned_t2e_vader
rule_base_scoring_long_df['vader_neg'] = VADER_score_list
rule_base_scoring_long_df.head()
```

	text_cleaned	vader_neg
0	Just another night. Another night of feeling l...	0.176
1	Is it possible to fake depression? I have been...	0.201
2	Imagine being attractive Imagine what it would...	0.144
3	Best moment to have anxiety It is am, I am tir...	0.128
4	hi, I am a year-old male from the uk, over the...	0.146

Figure 41: VADER on the Reddit dataset

Text2Emotion

For text2emotion, only the angry, surprise, fear and sadness score is considered (the happy score is omitted). Figure 42 and 43 shows the code snippet of how the sentences are passed into the text2emotion library functions and how the values are being appended to the dataframe for the Twitter and Reddit dataset respectively.

```
# create a list to store all emotion values from t2e for the short text
t2e_score_list_angry_short = []
t2e_score_list_surprise_short = []
t2e_score_list_fear_short = []
t2e_score_list_sadness_short = []

# do the t2e analysis
for sentence in short_data.text_cleaned_t2e_vader.tolist():
    t2e_score = te.get_emotion(sentence)
    t2e_score_list_angry_short.append(t2e_score['Angry'])
    t2e_score_list_surprise_short.append(t2e_score['Surprise'])
    t2e_score_list_fear_short.append(t2e_score['Fear'])
    t2e_score_list_sadness_short.append(t2e_score['Sad'])
```

```
# append the columns of scores to the existing dataframe
rule_base_scoring_short_df['angry_score'] = t2e_score_list_angry_short
rule_base_scoring_short_df['surprise_score'] = t2e_score_list_surprise_short
rule_base_scoring_short_df['fear_score'] = t2e_score_list_fear_short
rule_base_scoring_short_df['sadness_score'] = t2e_score_list_sadness_short
rule_base_scoring_short_df.head()
```

	text_cleaned	vader_neg	angry_score	surprise_score	fear_score	sadness_score
0	I get to spend New Year is home again alone an...	0.306	0.0	0.0	0.00	1.0
1	Depressed and lonely Stuck in a deep, never en...	0.527	0.0	0.0	0.40	0.6
2	Learning to pretend to have a good time had be...	0.054	0.0	0.0	0.33	0.0
3	So far he stop texting me after I said somethi...	0.103	0.0	0.5	0.50	0.0
4	sigh ?? I have not cried so much I am in so mu...	0.216	0.0	0.0	0.00	1.0

Figure 42: text2emotion on the Twitter dataset

```
# create a list to store all emotion values from t2e for the long text
t2e_score_list_angry_long = []
t2e_score_list_surprise_long = []
t2e_score_list_fear_long = []
t2e_score_list_sadness_long = []

# do the t2e analysis
for sentence in long_data.text_cleaned_t2e_vader.tolist():
    t2e_score = te.get_emotion(sentence)
    t2e_score_list_angry_long.append(t2e_score['Angry'])
    t2e_score_list_surprise_long.append(t2e_score['Surprise'])
    t2e_score_list_fear_long.append(t2e_score['Fear'])
    t2e_score_list_sadness_long.append(t2e_score['Sad'])
```

```
# append the columns of scores to the existing dataframe
rule_base_scoring_long_df['angry_score'] = t2e_score_list_angry_long
rule_base_scoring_long_df['surprise_score'] = t2e_score_list_surprise_long
rule_base_scoring_long_df['fear_score'] = t2e_score_list_fear_long
rule_base_scoring_long_df['sadness_score'] = t2e_score_list_sadness_long
rule_base_scoring_long_df.head()
```

	text_cleaned	vader_neg	angry_score	surprise_score	fear_score	sadness_score
0	Just another night. Another night of feeling l...	0.176	0.00	0.12	0.44	0.38
1	Is it possible to fake depression? I have been...	0.201	0.00	0.11	0.27	0.50
2	Imagine being attractive Imagine what it would...	0.144	0.00	0.23	0.15	0.46
3	Best moment to have anxiety It is am, I am tir...	0.128	0.05	0.32	0.34	0.15
4	hi, I am a year-old male from the uk, over the...	0.146	0.00	0.00	0.36	0.57

Figure 43: text2emotion on the Reddit dataset

WASSA 2017 EmoInt data

In this regression task, there will be three datasets being experimented and are being taken as the train and validation set, they are:

- WASSA 2017 Anger Dataset
- WASSA 2017 Fear Dataset
- WASSA 2017 Sadness Dataset

Then, the two depression datasets will be the test set and will get the predicted emotion score for each of the emotions. The two datasets are:

- Twitter Dataset
- Reddit Dataset

The models experimented are:

- Unidirectional LSTM
- Bidirectional LSTM
- Unidirectional GRU
- Bidirectional GRU

A total of 12 models will be built to determine the best performing one for each of the emotions (anger, fear and sadness) based on metrics like the RMSE and the MAPE.

The hyperparameters were experimented and these are the hyperparameters that produces the best performances in general:

- Number of Epochs: 20
- Learning Rate: 1e-04
- Hidden Dimension: 256
- Number of RNN layers: 2
- Dropout: 0.1
- Batch Size: 32
- Optimiser: Adam

Similar to the classification task elaborated in chapter 6, the anger, fear and sadness dataset will pass through the data loaders and then to build the sequence models.

After loading the datasets and building the models, the models are trained with the train data and evaluated on the validation data for each of the three datasets. The original WASSA 2017 EmoInt competition has already split the datasets into its corresponding train, validation and test set all with intensity score. Hence, in this main task, the train and test set of each emotion will be merged and become the new train set, while the validation set stays the same. Table 16, 17, 18 shows the model performances based on various metrics for the anger, fear and sadness dataset.

WASSA Anger Dataset

Model	MSE (Train Loss)	MAE	MSE	RMSE	MAPE (%)
Uni LSTM	0.02928	0.11075	0.01916	0.13841	27.011
Bi LSTM	0.03441	0.11089	0.01862	0.13646	25.407
Uni GRU	0.03719	0.10945	0.01927	0.13883	27.384
Bi GRU	0.03021	0.11617	0.02173	0.1474	28.224

Table 16: Model performance for the WASSA anger dataset

WASSA Fear Dataset

Model	MSE (Train Loss)	MAE	MSE	RMSE	MAPE (%)
Uni LSTM	0.03877	0.12445	0.02398	0.15485	28.928
Bi LSTM	0.03765	0.11534	0.02061	0.14357	26.902
Uni GRU	0.03244	0.12097	0.02234	0.14946	27.235
Bi GRU	0.03852	0.11806	0.02212	0.14873	27.773

Table 17: Model performance for the WASSA fear dataset

WASSA Sadness Dataset

Model	MSE (Train Loss)	MAE	MSE	RMSE	MAPE (%)
Uni LSTM	0.0378	0.11948	0.02409	0.15522	31.409
Bi LSTM	0.03841	0.10919	0.02154	0.14667	28.13
Uni GRU	0.04393	0.11725	0.02329	0.15261	30.006
Bi GRU	0.0403	0.1179	0.02406	0.15511	30.11

Table 18: Model performance for the WASSA sadness dataset

From Table 16, Table 17, Table 18, the bidirectional LSTM model performs the best for all the datasets in terms of the lowest RMSE loss. As such, the bidirectional LSTM models on the three datasets will be used for inference on the two test sets, which are the Twitter dataset and the Reddit dataset. RMSE is used as the main metric here because it gives a relatively high weight to large errors. Hence, RMSE is most useful when large errors are particularly undesirable [35].

As such, the test sets (Twitter and Reddit dataset) will be used for inference based on the three trained bidirectional LSTM models to obtain emotion scores for anger, fear and sadness. Table 19 and Table 20 shows five samples of the text entries with the three predicted emotion scores for the Twitter and Reddit dataset respectively.

Text	Fear Score	Angry Score	Sadness Score
Over the past 2 days I've drank 27 bottles of beer, 2 pints of cider and half a bottle of prosecco.	0.433326453	0.428122222	0.469055682
I'm having trouble catching my breath.	0.509710968	0.484118462	0.403120786
Hello darkness my old friend I've come to talk with you again...	0.364733905	0.439390928	0.4998959
I think my habit of reading in the morning is the root cause of the feeling throughout the day! specially in 2016	0.502374768	0.311957717	0.517002344
Mentally suffered #iwanttodie #worthless #lifewithoutcolor #pain #suicidal	0.794640601	0.454618484	0.796416521

Table 19: Five text entries from the Twitter dataset and the three predicted emotion scores

Text	Fear Score	Angry Score	Sadness Score
I'm not good at anything I used to draw good but now i've gotten worse at it for some reason. I have no idea how life works, i suck at socializing and keeping up relationships. Even if i invest 300 hours in a game i am still below average at it. The fact that i am constantly tired and have 0 energy doesn't help me much either. No matter what i do i am never more than average at it.	0.245335	0.27027	0.359572

After struggling more than a year I finally cut off all contact with my crush I deleted her from every friendlist, unfollowed her everywhere and basically removed her from my life after years of friendship. I just couldn't deal with it anymore. Loving her is one of the reasons I am as miserable as I am right now and as stupid as it sounds, one of the reasons I think about suicide daily just because I can't get her out of my head. I hope it helps and I hope she'll never get in touch with me and asks what's going on. I just want to forget her.	0.462129	0.476592	0.530932
Don't want to live, don't want to die. I just want to stop existing. Vanish into thin air without a trace and wipe my entire existence off the face of the planet. I don't want anyone to love me or miss or care about me. I don't remember the last time I was actually happy and I don't see things ever getting better because they've been bad for so long. Nothing ever changes no matter how hard I try. I can't stand being alone, but when I'm around anyone I just want to isolate. I can't eat because I get sick after 2 bites. I can't sleep because I have nightmares or always dream about undesirable things. I just don't know anymore. This isn't living, but whatever it is, I don't want to do it.	0.406061	0.393941	0.442113
"Life has no meaning the moment you lose the illusion of being eternal." This quote screwed me up. I have been constantly struggling to find meaning in my life and then I come across this quote and it absolutely drains me. It is validation for my worries and pain. I just want this to be over.	0.528386	0.396971	0.544874
There are so many things I want to say to people, things I want to type... But in the end, no matter how much of that text I wrote or how much I think about what to say, I delete it in the end. I'm sure I'm not alone in this regard, but I feel like 90% of all potential comments and posts on reddit I make are deleted before I post them. "Somebody else probably said it already." "Nobody cares what I have to say anyway." Just things like that. Lots more too, but I'm just going to post this now while I still have the nerve to do it.	0.483152	0.405836	0.506131

Table 20: Five text entries from the Reddit dataset and the three predicted emotion scores

Combining the scores together

To combine the scores, simply compute the individual scores to multiply with the weightage defined in section 7.1.1 above to produce the final depression score for all the entries in both the Twitter and the Reddit dataset. Figure 44 shows a code snippet of how the computations are done.

```
# get the depression metric score based on the above equation

# short dataset
depression_score_short_list = []
for i in tqdm(range(len(short_score_df))):
    dep_score = 0.2*short_score_df.vader_neg[i] + \
                0.05*(short_score_df.anger_score_t2e[i]+short_score_df.surprise_score_t2e[i]+short_score_df.fear_score_t2e[i]) + \
                0.25*(short_score_df.sadness_score_t2e[i]) + \
                0.05*(short_score_df.anger_score_pred[i]+short_score_df.fear_score_pred[i]) + \
                0.30*(short_score_df.sadness_score_pred[i])
    depression_score_short_list.append(dep_score)

# long dataset
depression_score_long_list = []
for i in tqdm(range(len(long_score_df))):
    dep_score = 0.2*long_score_df.vader_neg[i] + \
                0.05*(long_score_df.anger_score_t2e[i]+long_score_df.surprise_score_t2e[i]+long_score_df.fear_score_t2e[i]) + \
                0.25*(long_score_df.sadness_score_t2e[i]) + \
                0.05*(long_score_df.anger_score_pred[i]+long_score_df.fear_score_pred[i]) + \
                0.30*(long_score_df.sadness_score_pred[i])
    depression_score_long_list.append(dep_score)
```

Figure 44: Combining the scores to produce the final depression score

Table 21 and Table 22 shows five samples of the text entries with the final depression score for the Twitter and Reddit dataset respectively.

Text	Depression Score
Over the past 2 days I've drank 27 bottles of beer, 2 pints of cider and half a bottle of prosecco.	0.308789138
I'm having trouble catching my breath.	0.382627707
Hello darkness my old friend I've come to talk with you again...	0.214775012
I think my habit of reading in the morning is the root cause of the feeling throughout the day! specially in 2016	0.228817327
Mentally suffered #iwanttodie #worthless #lifewithoutcolor #pain #suicidal	0.715787911

Table 21: Five text entries from the Twitter dataset and the final depression score

Text	Depression Score
I'm not good at anything I used to draw good but now i've gotten worse at it for some reason. I have no idea how life works, i suck at socializing and keeping up relationships. Even if i invest 300 hours in a game i am still below average at it. The fact that i am constantly tired and have 0 energy doesn't help me much either. No matter what i do i am never more than average at it.	0.353994214
After struggling more than a year I finally cut off all contact with my crush I deleted her from every friendlist, unfollowed her everywhere and basically removed her from my life after years of friendship. I just couldn't deal with it anymore. Loving her is one of the reasons I am as miserable as I am right now and as stupid as it sounds, one of the reasons I think about suicide daily just because I can't get her out of my head. I hope it helps and I hope she'll never get in touch with me and asks what's going on. I just want to forget her.	0.457991443
Don't want to live, don't want to die. I just want to stop existing. Vanish into thin air without a trace and wipe my entire existence off the face of the planet. I don't want anyone to love me or miss or care about me. I don't remember the last time I was actually happy and I don't see things ever getting better because they've been bad for so long. Nothing ever changes no matter how hard I try. I can't stand being alone, but when I'm around anyone I just want to isolate. I can't eat because I get sick after 2 bites. I can't sleep because I have nightmares or always dream about undesirable things. I just don't know anymore. This isn't living, but whatever it is, I don't want to do it.	0.348622661
"Life has no meaning the moment you lose the illusion of being eternal." This quote screwed me up. I have been constantly struggling to find meaning in my life and then I come across this quote and it absolutely drains me. It is validation for my worries and pain. I just want this to be over.	0.376382664
There are so many things I want to say to people, things I want to type... But in the end, no matter how much of that text I wrote or how much I think about what to say, I delete it in the end. I'm sure I'm not alone in this regard, but I feel like 90% of all potential comments and posts on reddit I make are deleted before I post them. "Somebody else probably said it already." "Nobody cares what I have to say anyway." Just things like that. Lots more too, but I'm just going to post this now while I still have the nerve to do it.	0.298704001

Table 22: Five text entries from the Reddit dataset and the final depression score

7.2 Method 2: Using Sentic API on Depression Identification

As compared to method 1, this method is more straightforward as only API calls are needed to get the depression score.

7.2.1 Implementation

The datasets used in this subsection will be the same as the ones used in method 1. The API takes in the Twitter and the Reddit dataset and then returns a depression score in terms of a percentage.

The API call format for depression identification is as follow:

https://sentic.net/api/en/{API_key_for_depression_identification}.py?text={your_text}

Figure 45 shows a code snippet of calling the API in python using the library 'requests', with a test sentence to observe the output depression score and the status code in case there are any errors.

```
# test sentence
response = requests.get(f"https://sentic.net/api/en/{DEPRESSION_IDENTIFICATION_KEY}.py?text=a little sad and happy somehow")
print(response.status_code)
# the values received as the response, usually is in json, but this is in string
print(response.text)
# check the type of the response
print(type(response.text))
```

```
200
33%

<class 'str'>
```

Figure 45: A sample call to Sentic API

From Figure 45, the response's status code is 200, which implies that the request of the API call has succeeded, and the depression score is returned. However, a string value of '33%' is returned as seen in the code snippet. There is a need to convert the string value of '33%' to a float decimal of 0.33, so that this value is standardised with the depression score obtained in method 1 for ease of comparison in later subsection. In addition, not all the text input for the API will succeed like in this example. Hence, Figure 46 shows a helper function which incorporates regular expressions to remove non-numerical values, such as '%'. Also, there is a conditional check to see if the response's status code is 200 to ensure that a depression score is returned from the API. If the status code is not 200, the score will be set to -1 to indicate that there is an error in that text entries and no response was returned. Figure 47 shows a text sample from the Twitter dataset being passed into the helper function.

```
def get_depression_score_from_api(text):
    response = requests.get(f"https://sentic.net/api/en/{DEPRESSION_IDENTIFICATION_KEY}.py?text={text}")
    # check status code if success or not
    if response.status_code == 200:
        score = float(re.sub("[^0-9.]", "", response.text))*0.01
    else:
        print("status code not 200!")
        score = -1 # negative one to denote that an error occurred when processing this piece of text

    return score
```

Figure 46: Helper function to call the API and also to pre-process the response

```
# try on one piece of data
score = get_depression_score_from_api(short_data.text_cleaned_t2e_vader[10])
score
```

```
0.466
```

Figure 47: A function call to the helper function

From Figure 47, the output is as expected in float decimal and without the '%'. The Twitter and Reddit dataset entries are then ready to pass into this helper function to get the depression scores for all the entries. After this process, the text entries with depression score -1 (error) are then transformed to zeros so that comparisons with the depression scores obtained in method 1 are numerically feasible.

Table 23 and Table 24 shows five samples of the text entries with the depression scores that are returned from the API for the Twitter and the Reddit dataset respectively.

Text	Depression Score
Over the past 2 days I've drank 27 bottles of beer, 2 pints of cider and half a bottle of prosecco.	0
I'm having trouble catching my breath.	0.33333
Hello darkness my old friend I've come to talk with you again...	0.22

I think my habit of reading in the morning is the root cause of the feeling throughout the day! specially in 2016	0.25
Mentally suffered #iwanttodie #worthless #lifewithoutcolor #pain #suicidal	0.332

Table 23: Five text entries from the Twitter dataset and their depression score from the Sentic API

Text	Depression Score
I'm not good at anything I used to draw good but now i've gotten worse at it for some reason. I have no idea how life works, i suck at socializing and keeping up relationships. Even if i invest 300 hours in a game i am still below average at it. The fact that i am constantly tired and have 0 energy doesn't help me much either. No matter what i do i am never more than average at it.	0.376
After struggling more than a year I finally cut off all contact with my crush I deleted her from every friendlist, unfollowed her everywhere and basically removed her from my life after years of friendship. I just couldn't deal with it anymore. Loving her is one of the reasons I am as miserable as I am right now and as stupid as it sounds, one of the reasons I think about suicide daily just because I can't get her out of my head. I hope it helps and I hope she'll never get in touch with me and asks what's going on. I just want to forget her.	0.4726
Don't want to live, don't want to die. I just want to stop existing. Vanish into thin air without a trace and wipe my entire existence off the face of the planet. I don't want anyone to love me or miss or care about me. I don't remember the last time I was actually happy and I don't see things ever getting better because they've been bad for so long. Nothing ever changes no matter how hard I try. I can't stand being alone, but when I'm around anyone I just want to isolate. I can't eat because I get sick after 2 bites. I can't sleep because I have nightmares or always dream about undesirable things. I just don't know anymore. This isn't living, but whatever it is, I don't want to do it.	0.66
"Life has no meaning the moment you lose the illusion of being eternal." This quote screwed me up. I have been constantly struggling to find meaning in my life and then I come across this quote and it absolutely drains me. It is validation for my worries and pain. I just want this to be over.	0.51067
There are so many things I want to say to people, things I want to type... But in the end, no matter how much of that text I wrote or how much I think about what to say, I delete it in the end. I'm sure I'm not alone in this regard, but I feel like 90% of all potential comments and posts on reddit I make are deleted before I post them. "Somebody else probably said it already." "Nobody cares what I have to say anyway." Just things like that. Lots more too, but I'm just going to post this now while I still have the nerve to do it.	0.1525

Table 24: Five text entries from the Reddit dataset and their depression score from the Sentic API

7.3 Depression score comparison between the two methods

Table 25 and Table 26 shows the comparison in the depression scores for five samples of texts for the Twitter and Reddit dataset respectively.

Text	Symbolic + Subsymbolic	Sentic API
so alone so tired so bored so ugly so depressed so so so so so so	0.66375359	0.866
Sleeping for 12+ hours at a time, and fantasizing about ways to kill myself.	0.306328882	0.33
Feeling bummed out rn. Family is disappointing and friends are too far away. ??	0.384605402	0.5
Shitty is the worst feeling ever #depressed #anxiety	0.697436402	0.8325
Why is it that every one gets what I want, I guess whatever...	0.219623731	0.33

Table 25: Comparison of depression score between the two methods for the Twitter dataset

Text	Symbolic + Subsymbolic	Sentic API
Why I always let myself fooled... Why I always think there is something more than there really is... I always think I'm good friends with someone and I act accordingly. I try to be that good friend. Then at some point I always realize that it's not a real friendship. The other one didn't think it that seriously. The other one just talked with me because at that time it was fun. It feels crushing. Realizing every single time that the person you took as a good friend, or even possible friend isn't really thinking same about you. Realizing how stupid, silly and naive you have been thinking that you could be a friend for that person. Realizing how they just are amused that you were so deep involved in it.	0.279123936	0.3525
help..? Has anyone reached a point in their life where they knew they needed help but wasnt quite sure how to ask for it or even know what it is they needed help with and who they would need to ask in order to figure out what they needed help? not really a depressive question but more of a question i felt might be better in the depression tread	0.30549448	0.19

I lost my father recently, and now I have less suicidal thoughts This isn't gonna be what you think it's gonna be. If you read the title you're probably thinking my father was the reason of my issues. He was not, I loved him very dearly. I am extremely sad he passed away while I'm in such a young age (I'm 21) but c'est la vie I guess.. The reason I'm now less likely to commit suicide is actually I won't / can't do this to my mother. We had to plan and pay for the funereal and oh my days it's expensive. Already we were on the line of poverty, and this didn't quite help. I just can't do it to my mother to spend even more money that she doesn't really have. So now the feeling of guilt 'if' I were to do it is bigger than my misery. Now I just slate the days away, it's hard, but there's no other option for me right now..	0.349676793	0.28826
extreme sadness whenever seeing anything happy Hello, this has been bothering me for a while and i have no idea what is this but when i ever i read or see anything happy like a happy scene or a happy story from some random comment i feel a huge wave of sadness wash over me and i feel like crying, why? what is going on? (i am not sure if this belongs to here but if not i don't mind it being deleted and sorry for breaking rules.)	0.269165297	0.38778
Why is life horrible to me and only me It just seems like one bad thing after another. Every time I try to reach out and open up to people it just goes down in flames. They always be like , no your not depressed , its your fault that your this way and only you can helped yourself. Always shunning me away and ignoring me. Then its always my fault. You didn't help with the decorations its your fault it failed.	0.414519921	0.64722

Table 26: Comparison of depression score between the two methods for the Reddit dataset

From Table 25 and Table 26, both depression scores for the sampled text entries are numerically quite close to each other.

7.4 Limitations in the Comparisons

Due to the time factor and the lack of annotators from relevant fields like Psychology or Linguistics to give a golden depression score for the individual entry in the dataset, there is no way to compare the losses and determine which method is better in producing the depression score of the text entries in the data.

7.5 Conclusion for this chapter on Emotion Intensity Prediction

In this chapter, there are two methods used to predict the depression scores for the Twitter and the Reddit data. One way is to use a combination of symbolic and sub-symbolic approaches to get the depression score. The other way is to use the Sentic API on depression identification to get the depression score. Then, there were limitations to the comparisons between these two methods as there was a lack of annotators who are experts in the field of depression in this project.

CHAPTER 8: SUBTASK - TEXT SUMMARISATION

Sometimes, depressive texts might be very lengthy and one only has a limited amount of time to skim through them. Relevant authorities might only be interested in the most important segment of the entire text as those texts will decide the level of depression of that social media user and hence, suitable measures will be implemented accordingly. As such, text summarisation is needed in order to retain only the most important segments of the entire text.

Text Summarisation is the technique for generating a concise and precise summary of large texts while focusing on the sections that convey useful information and retaining its overall meaning. There are two approaches towards Text Summarisation, Extractive Summarisation and Abstractive Summarisation. Extractive Summarisation is an approach that identifies the important sentences or phrases from the original text and extracts only those from the text. There will not be new sentences formed. Abstractive Summarisation on the other hand generates new sentences from the original texts. The new sentences may not be present in the original text [49].

This chapter will look at both Extractive and Abstractive Summarisation in the context of depressive texts. Likewise in this chapter, the same Twitter and Reddit dataset will be analysed.

8.1 Brief Outline

In this project the two summarisation techniques are as follows:

- **Extractive Text Summarisation**
  - DeepAI Text Summarisation API

- **Abstractive Text Summarisation**
  - Pre-training with Extracted Gap-sentences for Abstractive Summarisation (PEGASUS)

8.2 Extractive Text Summarisation

The DeepAI Text Summarization API call format using the python library 'requests' for extractive text summarisation is as follows:

```
requests.post(
    "https://api.deepai.org/api/summarization",
    data={
        'text': 'YOUR_TEXT_DATA_HERE',
    },
    headers={'api-key': API_KEY}
)
```

8.2.1 Implementation

The API will take in the text input from the Twitter or Reddit data entries and the API will return the summarised text. A helper function is written to ease the process when the whole dataset is being passed into the API. Figure 48 shows the code snippet of the helper function.

```
def get_summarized_text(text):
    r = requests.post(
        "https://api.deepai.org/api/summarization",
        data={
            'text': text,
        },
        headers={'api-key': API_KEY}
    )
    text = r.json()['output']
    text = text if text != '' else '-'
    return text
```

Figure 48: Helper function for the Extractive Text Summarisation API call

Not all the text can be summarised. Especially for relatively shorter text in the Twitter dataset, there are quite a number of text entries who do not have a summary. Table 27 and Table 28 shows a few samples of the raw texts and its corresponding summarised texts for the Twitter and the Reddit dataset respectively.

Text	Summarised Text
I did not want to wake up. I was having a much better time asleep. And that is really sad.	-
I walk around the house and I see things of Debbie and I think of her. I think of the pain she went through, suffered for so long. All those procedures she went through. Between finding out she had cancer and the pain. Having chemotherapy for a long time.	I think of the pain she went through, suffered for so long.
I am always uncomfortable, especially now that I have picked up so much weight. I am the heaviest I have ever been in my life. .	-
And I am starting to break down;	-
one sec ur and the other ur	-
Woke up feeling AF. Okay. Guess that is how today is gonna be.	-
I do not think anyone fully comprehends how my back pain affects me emotionally.	-
Never looking for pity. Just looking for company and consolation.	-
Depression reality: Trapped inside my skin and trying to tear myself out but no-one knows I am trapped or sees my struggle.	-
Hello darkness my old friend	-
I feel like I am well-liked as well. I mean, I seriously doubt anyone would describe me as anything but kind.	-
gelo.. abis baca ini.. sedih jg yaahh.. it is a very familiar sad feeling and i do not want to be this kind of mother but sometimes i cannot help but feel like i am falling into an inescapable toxic cycle.. i just.. want to cry.. ??	it is a very familiar sad feeling and i do not want to be this kind of mother but sometimes i cannot help but feel like i am falling into an inescapable toxic cycle..
have not been on Twitter all day.	-

Table 27: Samples of the raw texts and the summarised texts for the Twitter dataset

Text	Summarised Text
No end in sight I have been depressed for so long. I just feel a void,dark and empty. I can feel the pain of no one loving me. I have zero friends. I live with toxic parents .I feel like i have been in an endless loop with no end in sight to the pain I am in. I hate my existence. I have done therapy,yoga blah blah. I still feel miserable. I am so unlucky. Ca not do anything right. What is the point if you are not good at anything. Im below average in life,no money,No job,No talent and no interest. I do not get it. Why am I even here?	I can feel the pain of no one loving me. I live with toxic parents .I feel like i have been in an endless loop with no end in sight to the pain I am in.
I am new here. I had been having a lot of problems with anxiety/depression and that was affecting my job a lot. I went out of work for a week hoping that maybe that would help. I felt like I was doing ok and that the relaxing was really helping but the night before I was supposed to go back to work it all happened again. So I pretty much came to the realization that I cannot do it without medicines. Is this normal to think you are ok and then have it happen again?? And also, have any of you had your jobs affected by it?	I had been having a lot of problems with anxiety/depression and that was affecting my job a lot.
Poem about mental illness I wrote When you ask if I am okay Of course I am okay. I am alive, But not quite living. I am okay but I am trapped, Watching those around me Live their lives. I am okay but I feel lost, Or rather I cannot feel at all. I am okay but I am screaming As loud as I can inside my head. I want to be heard But suffer in silence. I am okay but other is are not I make them feel helpless As they watch me sway Between okay and isolation. Isolation from others From myself From my passions. Isolated from my own feelings Sometimes there is nothing left but the dark. You feel helpless? I am helpless I am drowning in the darkness. I am okay But I am not really here Here or there I am everywhere My brain cannot straighten up. I am okay but I am not living I feel trapped within myself. I am not okay.	I am okay but other is are not I make them feel helpless As they watch me sway Between okay and isolation. I am okay but I am not living I feel trapped within myself.

Table 28: Samples of the raw texts and the summarised texts for the Reddit dataset

As seen in Table 27, the shorter Twitter dataset does not have a summary as the raw text entries are short. Only those with longer text have a summary. From Table 28, it is evident that the summarised sentences from the Reddit dataset retained its text structure from the raw sentences, implying that there were no new sentences formed during summarisation.

8.3 Abstractive Text Summarisation

Unlike the extractive text summarisation, abstractive text summarisation generates new sentences from the raw texts, implying that the summarised text might not appear in their corresponding raw text. Pegasus will be used to conduct the abstractive text summarisation.

8.3.1 PEGASUS

PEGASUS is a state-of-the-art transformer model for abstractive text summarisation and uses an encoder-decoder model for sequence-to-sequence learning. The encoder will first consider the context of the whole input text and encode it into a context vector, which is just a numerical representation of the input text. Then, this numerical representation will pass through the decoder to decode the context vector to produce the summary [50][51].

8.3.2 Implementation

The Hugging Face's transformers library will be used to import PEGASUS. There are a variety of datasets that PEGASUS are trained on. Figure 49 shows the variety of datasets in the models page from the Hugging Face website [52].

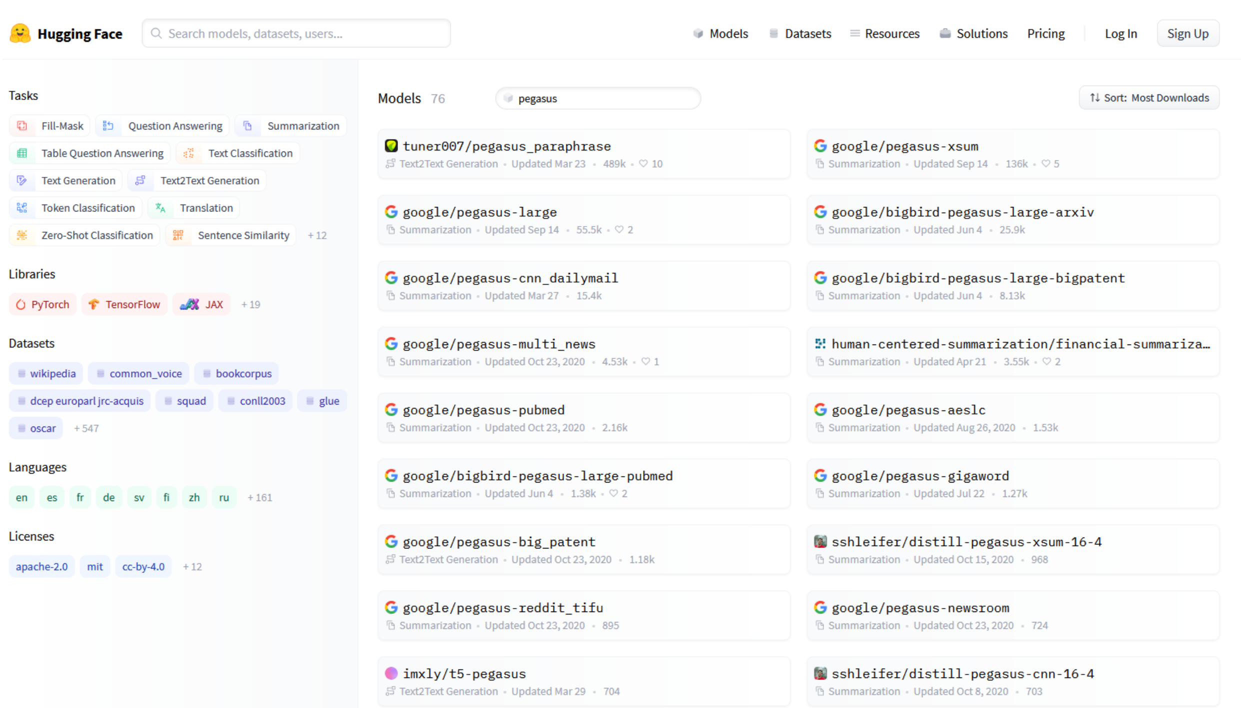

Figure 49: PEGASUS trained on different datasets

In this project, the pretrained models trained from the pegasus-xsum and pegasus-reddit_tifu will be used to summarise the depressive Twitter and the Reddit dataset because they are the more commonly used pretrained models. Figure 50 shows a code snippet to load the pretrained models and its tokenizers. Figure 51 shows a code snippet of the helper function to load the text from the Twitter and Reddit dataset, then pass it to the pretrained model for summarisation.

```
# load tokenizers
tokenizer_xsum = PegasusTokenizer.from_pretrained("google/pegasus-xsum")
tokenizer_reddit = PegasusTokenizer.from_pretrained("google/pegasus-reddit_tifu")

# load the models
model_xsum = PegasusForConditionalGeneration.from_pretrained("google/pegasus-xsum").to(device)
model_reddit = PegasusForConditionalGeneration.from_pretrained("google/pegasus-reddit_tifu").to(device)
```

```
Downloading: 100% 1.91M/1.91M [00:01<00:00, 1.88MB/s]
Downloading: 100% 65.0/65.0 [00:00<00:00, 1.26kB/s]
Downloading: 100% 87.0/87.0 [00:00<00:00, 2.06kB/s]
Downloading: 100% 3.52M/3.52M [00:01<00:00, 3.44MB/s]
Downloading: 100% 1.36k/1.36k [00:00<00:00, 33.1kB/s]
Downloading: 100% 1.91M/1.91M [00:01<00:00, 2.07MB/s]
Downloading: 100% 65.0/65.0 [00:00<00:00, 1.72kB/s]
Downloading: 100% 87.0/87.0 [00:00<00:00, 2.11kB/s]
Downloading: 100% 1.12k/1.12k [00:00<00:00, 25.8kB/s]
Downloading: 100% 2.28G/2.28G [01:11<00:00, 34.9MB/s]
Downloading: 100% 2.28G/2.28G [01:48<00:00, 15.1MB/s]
```

Figure 50: Loading the pretrained PEGASUS model and its tokenizer

```
# helper function to summarize the text
def summarize(text, tokenizer_type, model_type):
    # create tokens - number representation of our text
    tokens = tokenizer_type(text, truncation=True, padding="longest", return_tensors="pt").to(device)

    # summarize the text
    summary = model_type.generate(**tokens)

    return tokenizer_type.decode(summary[0])
```

Figure 51: Helper function to summarise the texts

Table 29 and Table 30 shows a few samples of the raw texts and its corresponding summarised texts for the Twitter and the Reddit dataset respectively.

Text	Summarised Text (xsum)	Summarised Text (reddit_tifu)
I am in tears. This is so heartbreaking	This is one of the most heartbreaking videos I have ever seen.	i am in tears.
Sucks that the only place I can go to if I cannot handle , is the one I do not feel safe in. And the only person I can _try_ to ask help if I need it, is the one I do not trust.	Is there a place in the world where I can ask for help if I need it, and the only person I can trust is the one I do not trust?	don't go to the one you don't trust if you can't handle it,
It is very hard to feel better because it is been so long that I do not remember what better felt like!!	It has been more than a year since I last wrote a blog.	it's been so long that it's been so hard to feel better because it's been so long that i do not remember what better felt like!!
She did not go out with anybody. She died. It is really hard to accept sometimes. I feel lonely. I feel lost. I feel like I do not have anybody there for me anymore.. who can pick up the slack, when I am lacking. I guess that I am glad she is not in pain. I sure miss her.	It is a year since my mother died.	i feel lonely because i do not have anybody to pick up the slack..
I want to give up. I want to go home. I feel like nothing and I will never heal.	I want to die.	i want to go home. i feel like nothing.

Table 29: Samples of the raw texts and the summarised texts for the Twitter dataset

Text	Summarised Text (xsum)	Summarised Text (reddit_tifu)
How is it going, all who read. My name is Greg, I am 19 and have had depression for basically as long as I remember. Here is a brief history of my journey through depression and the treatments I have endeavored. I began on lexapro. It seemed to be helping and then it quit on me (or perhaps vice versa). After that I tried zoloft, and wellbutrin, but to the same effect. Then I turned my world to narcotics (mainly, marijuana and DXM). They helped ALOT, but because they are illegal, I felt a constant public stigma and guilt. I could not take it, despite the amazing effects of both drugs. I am now taking Paxil CR, but I am going to quit it because it is not working. I am beginning to lose hope altogether. It seems apparent to me that SSRIs just are not for me, or maybe I just have not found the right one. The real problem with depression is that you have to wait a month on each drug just to see it it is going to work. Those months can feel like years. Anyways, that is who I am. I hope to make a few friends here, and a lot of advice as well. Does anyone one else share these symptoms (if so, and if you think you can help contact me please): Chronic fatigue, lack of motivation/desire, inability to enjoy life, insomnia (including the other obvious and supplementing symptoms) The real killer is I know I have the potential to be someone great, but I have to become happy first..Greg	My name is Greg, I am 19 and have had depression for basically as long as I remember.	i have had depression for as long as i can remember, and have tried a lot of drugs to help, but have not been able to find the right one.
Ca not relate or make friends with anyone my age 19 years old, just got out of highschool. Anyways, I am a chill dude, good looking, and sociable. Anyways, I have trouble making any friends with dudes my age (no trouble with girls). I will vibe with them really well at first and play sports or whatever with a few of them. They seem pretty friendly but at the same time some of their other friends seem to kind of hate me even though they act polite when interacting with me. I can even hear them talk shit sometimes from a distance even tho there is no personal beef and most of them have not spoken more than a sentence to me b4? however they will not directly confront me and at least be upfront. Literally even in highschool I would have dudes try to test me and start shit with me though I didnt even know them lol this shit, is every kid my age a bad boy? I met a few chill ppl, but damn most of them seem like insecure cunts..it is just making me hate ppl more and myself more.	I'm 19 years old and I have trouble making friends with anyone my age.	got out of highschool, have trouble making friends with dudes my age, and some of them seem to hate me even though they act nice when interacting with me.
drugs are the only thing that make me feel okay i have done	Drugs are the only	drugs are the only thing

everything under the sun to try and help myself. counseling, confiding in anyone who will listen, taken all different types of medication, and the only thing that has ever made me feel okay is not being sober. idc what it is i will do it. i hate being sober. i just want to get screwed up and lay in bed and watch shit on my laptop and i literally have no reason to be so depressed but the only thing that helps is drugs. not looking for moral support or anything i just need to vent without being looked down on and judged. thx	thing that make me feel okay i have done everything under the sun to try and help myself, confiding in anyone who will listen, taking all different types of medication, and the only thing that has ever made me feel okay is not being sober.	that make me feel okay i have done everything under the sun to try and help myself. drugs are the only thing that make me feel okay i have done everything under the sun to try and help myself. drugs are the only thing that make me feel okay i have done everything under the sun to try and help myself.
Maybe They are Wrong About It All We are always told "when you are depressed, your brain is lying to you about how bad everything is". What if they are wrong? If someone is piss poor and starving, cannot pay their bills, replace their moth-eaten clothes or go out to do interesting things, surely their money situation really IS as bad as all that. If someone has no family, no friends and has been bullied/abused by everyone they have met in life, perhaps their social situation really IS as bad as all that. If someone has had lots of physical health issues like cancer, chronic pain, obesity etc, maybe their physical woes really ARE as bad as all that. Some people go through all of the above, all at once. So maybe life IS as bad as all that for them. The whole "your brain is lying about how bad things are" thing often just feels like "you are just imagining your problems". Like, I know depression etc make your brain incapable of dealing with them properly, but it does not mean legitimate problems do not exist and that some people is lives are not thoroughly worthless all the same.	Have you ever heard someone say "life is as bad as depression"?	"your brain is lying to you about how bad everything is" does not mean everything is as bad as you think it is, just that it is not as bad as you think it is.
Nice to see you too!! How are you today? I think you really have hit the mark on what I was trying to express with this question - all the negativity and darkness that is contained in almost every media outlet you can name. The disasters you mention, such real human tragedy but the way the media went after it was appalling and seemed to be muddled in politics and ratings. As for those forensic shows - ugh! I used to be able to watch them but way before this recent "episode" of mine I stopped. I could not stand the heartache of knowing what evil lurks out there. Where I work now one of the first friends I made was a gentleman who has since retired. He told me that he did not read the news, watch TV or listen to the radio (except for light jazz) because he found that to do so was	What do you think the media do for us?	don't watch forensic shows, read the news, listen to the radio, or watch the news at all - it will make you feel like shit.

just too much pain, it did not "serve" him to subject himself to that kind of pain. He seemed to be generally very happy and upbeat to me so there must be something to it. As for your friend - that is a tough one but maybe since she is your friend she might understand if you explain to her how incredibly painful her TV choices are on you. Does she ever read or post here? Anyway, it is GREAT talking with you again and I look forward to the next chat. I hope you have a wonderful day and weekend.		

Table 30: Samples of the raw texts and the summarised texts for the Reddit dataset

8.4 Conclusion for this chapter on Text Summarisation

In this chapter, there are two ways to summarise a piece of text. One way is the extractive text summarisation where the structure of the sentences is preserved and no new sentences are formed. The other way is the abstractive text summarisation where the structure of the sentences changes and new sentences might be formed.

CHAPTER 9: MAIN TASK 3 - EMOTION-CAUSE PAIR EXTRACTION

This chapter will go a step deeper from emotion classification and emotion intensity prediction, by looking at the emotion 'depression' and the likely cause for it. Then, it extracts the emotion and the cause pair accurately. There is a need to determine the cause of depression as different symptoms of depression require different treatments to suppress it.

9.1 Previous research on Emotion-Cause Pair Extraction (ECPE)

This chapter is inspired by the research paper on "An End-to-End Network for Emotion-Cause Pair Extraction" [53]. ECPE aims to extract all the potential clause-pairs of emotions and their corresponding causes in a document. The authors also state that previous ECPE tasks either followed a multi-stage approach where the emotion, cause and the pairing are done separately or the use of complex architectures to resolve the limitations. However, the authors will be adopting the end-to-end model approach to demonstrate the effectiveness of joint training on the ECPE task. The authors use the benchmark NTCIR-13 workshop dataset in their experiment [54]. In the benchmark dataset, six emotions were explored: disgust, fear, anger, happiness, sadness, and surprise. An example paragraph separated into individual clauses for analysis is shown below.

Clause 1: Adele arrived at her apartment late in the afternoon after a long day of work.
Clause 2: She was still furious with her husband for not remembering her 40th birthday.
Clause 3: As soon as she unlocked the door, she gasped with surprise;
Clause 4: Mikhael and Harriet had organised a huge party for her.

The above example contains two emotion-cause pairs. Clause 2 is an emotion clause (anger) and is also the corresponding cause clause (for not remembering her 40th birthday). Clause 3 is an emotion clause (surprise) and Clause 4 is its corresponding cause clause (organised a huge party for her).

9.1.1 Task Definition

Formally, a document consists of text that is segmented into an ordered set of clauses D = [$c_1, c_2, \ldots, c_d$] and the ECPE task aims to extract a set of emotion-cause pairs P = {..., (c_i, c_j), …} ($c_i, c_j \in$ D), where c_i is an emotion clause and c_j is the corresponding cause clause [53]. Two auxiliary tasks are proposed as well, they are: Emotion Detection and Cause Detection

9.1.2 Approach

They propose models to take an entire document as an input and compute for each ordered pair of clauses (c_i, c_j), the probability of being a potential emotion-cause pair. They also suggest a hierarchical architecture. Word level representations are used to obtain clausal representations using bidirectional LSTM. The clause embedding is then fed into two separate clause-level encoders using bidirectional LSTMs (Emotion-Encoder and Cause Encoder), which corresponds to the two auxiliary tasks [53]. There are two ECPE variants proposed in this paper, they are:

- E2E-PExt(E)
    - The clause embedding will first be fed into the Emotion-Encoder to obtain the contextualised clause representation for the emotion and the predicted emotion.
    - Then, the predicted emotion, together with the clause embedding, is fed into the Cause-Encoder to obtain the contextualised clause representation for the cause and the predicted cause.
    - Lastly, the Cartesian product of the contextualised clauses will be taken and fed to a pair predictor which is a fully connected layer to produce the final emotion-cause pair prediction.
- E2E-PExt(C)
    - The clause embedding will first be fed into the Cause-Encoder to obtain the contextualised clause representation for the cause and the predicted cause.
    - Then, the predicted cause, together with the clause embedding, is fed into the Emotion-Encoder to obtain the contextualised clause representation for the emotion and the predicted emotion.
    - Lastly, the Cartesian product of the contextualised clauses will be taken and fed to a pair predictor which is a fully connected layer to produce the final emotion-cause pair prediction.

9.1.3 Diagrams of the Network Proposed

Figure 52 and Figure 53 show the architectures of the end-to-end network for ECPE for the two variants, E2E-PExt(E) and E2E-PExt(C) respectively [53].

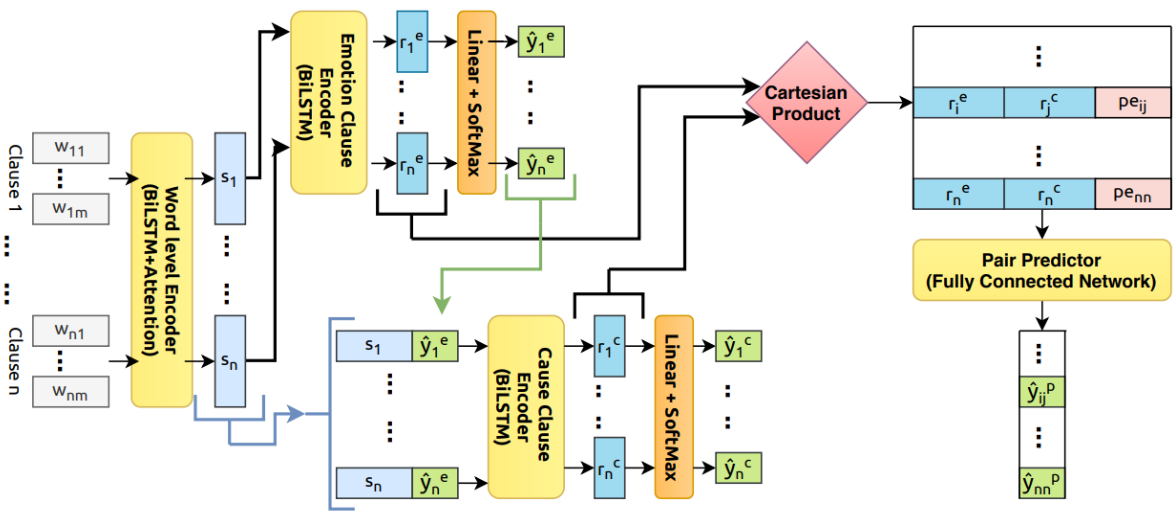

Figure 52: Architecture of E2E-PExt(E)

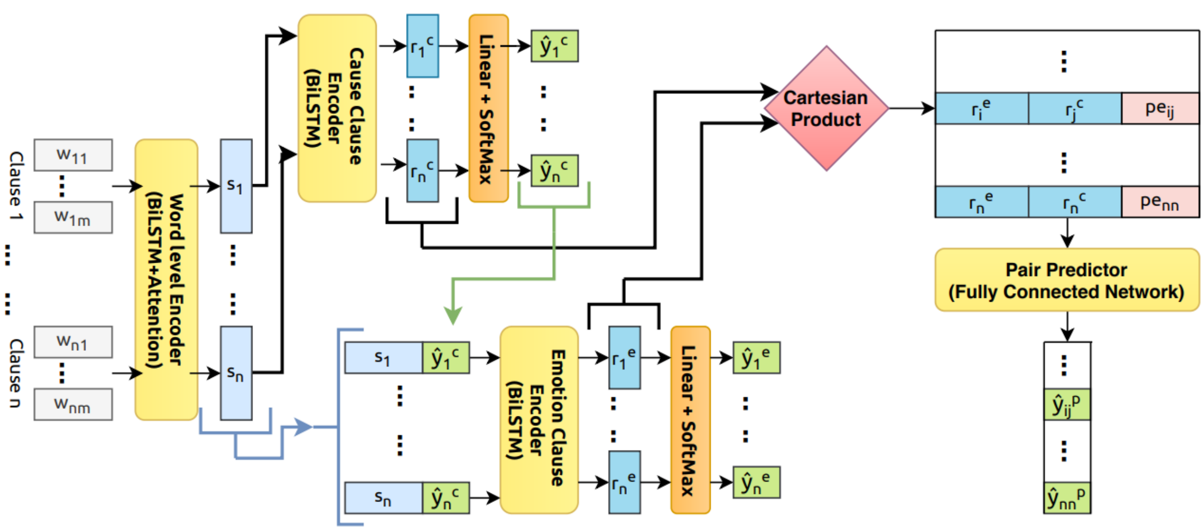

Figure 53: Architecture of E2E-PExt(C)

9.1.4 Performance on the NTCIR-13 workshop dataset

The authors have also stated the performances of the two ECPE model variants shown in Table 31 below.

Models	Emotion Extraction (%)			Cause Extraction (%)			Pair Extraction (%)		
	Precision	Recall	F1 Score	Precision	Recall	F1 Score	Precision	Recall	F1 Score
E2E-PExt(E)	71.63	67.49	69.43	66.36	43.75	52.26	51.34	49.29	**50.17**
E2E-PExt(C)	71.70	66.77	69.10	63.75	42.50	50.42	48.88	48.22	**48.37**

Table 31: Model Performances of the two ECPE model variants on the benchmark data

9.2 ECPE in the context of depression

The main goal of this chapter is to perform the ECPE task stated in section 9.1 but in the context of depression, hoping to obtain a similar F1 score for the pair extraction. The architecture remains mostly the same but the hyperparameters will be tuned again because the datasets used are different. The datasets used for this task will be the Twitter and Reddit dataset, but further pre-processing of these datasets is needed.

9.3 Further pre-processing of the datasets

Both the texts in Twitter and the Reddit dataset are in full sentences in the CSV format. There is a need to pre-process the dataset such that the data is presented in separate clauses for the ease of annotation of the emotion and the cause clauses. Figure 54 shows the code snippet of a helper function to separate the individual text entries into separate clauses.

```
def get_separate_clauses(text):
    doc = en(text)
    #deplacy.render(doc)

    seen = set() # keep track of covered words

    chunks = []
    for sent in doc.sents:
        heads = [cc for cc in sent.root.children if cc.dep_ == 'conj']

        for head in heads:
            words = [ww for ww in head.subtree]
            for word in words:
                seen.add(word)
            chunk = (' '.join([ww.text for ww in words]))
            chunks.append( (head.i, chunk) )

        unseen = [ww for ww in sent if ww not in seen]
        chunk = ' '.join([ww.text for ww in unseen])
        chunks.append( (sent.root.i, chunk) )

    chunks = sorted(chunks, key=lambda x: x[0])
    chunk_list = []

    for ii, chunk in chunks:
        chunk_list.append(chunk)

    return chunk_list
```

Figure 54: Code snippet of a helper function to separate texts into its individual clauses

Then, the clauses are appended into a text file and pre-process to make it the same format as the NTCIR-13 workshop dataset. Figure 55 shows the code snippet to append the clauses to the text file for the Twitter dataset. For the Reddit dataset, there exists a text entry with more than 200 clauses, which makes it computationally expensive for the model to train later. As such, the Reddit dataset will be summarised using the extractive text summarisation technique elaborated in chapter 8.2, before being passed into the model. Then, only text entries with at most five clauses are allowed to be written into the text file which is the final dataset for the ECPE task. Figure 56 shows the code snippet to append the clauses for the Reddit dataset. Figure 57 and Figure 58 shows a sample of the data in the text file for the Twitter and the Reddit dataset respectively. For the remaining analysis, the formatted dataset (text file format) will be used instead of the CSV file.

```
# split all the text data into clauses

# open text file , save as ecpe_raw for further annotations later
f = open(f'{DATA_PATH}ecpe_raw_short.txt','a+')

for i in tqdm(range(len(ecpe_data_short_clean))):
    chunk_list = get_separate_clauses(ecpe_data_short_clean.text_cleaned_ecpe[i])

    # do a regex here to remove the punctuations
    chunk_list_no_punct = []
    for chunk_str in chunk_list:
        out = chunk_str.translate(str.maketrans('', '', string.punctuation))
        chunk_list_no_punct.append(out)

    # enter data into the text file according to the format we want
    counter = i+1
    f.write(f'{counter} {len(chunk_list)}\n')
    f.write(f' (0, 0),\n')
    for index, clause in enumerate(chunk_list_no_punct):
        f.write(f'{index+1},null,null,{clause}\n')

# close text file
f.close()
```

Figure 55: Code snippet to append the clauses into the text file for the Twitter dataset

```
# split all the text data into clauses

# added a counter to see how many data are being stored after those conditions
data_counter = 0

# open text file , save as ecpe_raw for further annotations later
f = open(f'{DATA_PATH}ecpe_raw_long_summarized_max_5_clauses.txt','a+')

for i in tqdm(range(len(ecpe_data_long_summarized_clean))):
    chunk_list = get_separate_clauses(ecpe_data_long_summarized_clean.text_summarized[i])

    # only save those entries with not more than 5 clauses
    if len(chunk_list) > 5:
        continue

    # do a regex here to remove the punctuations
    chunk_list_no_punct = []
    for chunk_str in chunk_list:
        out = chunk_str.translate(str.maketrans('', '', string.punctuation))
        chunk_list_no_punct.append(out)

    # enter data into the text file according to the format we want
    counter = i+1
    data_counter += 1
    f.write(f'{counter} {len(chunk_list)}\n')
    f.write(f' (0, 0),\n')
    for index, clause in enumerate(chunk_list_no_punct):
        f.write(f'{index+1},null,null,{clause}\n')

# close text file
f.close()
```

Figure 56: Code snippet to append the clauses into the text file for the Reddit dataset

ecpe_raw_short.txt

```
22 3
 (0, 0),
1,null,null,When your scares get re opened an its like pooring salt in them
2,null,null,I hate this feeling
3,null,null,All the pain i m in again
23 4
 (0, 0),
1,null,null,Here it is  120am  and
2,null,null,I am wide awake
3,null,null,I thought was supposed to know my ass out
4,null,null,Instead my mind is racing
24 6
 (0, 0),
1,null,null,She would have some understanding
2,null,null,But noone  noone is there
3,null,null,People have their own lives and their own busy schedules
4,null,null,I have nothing but time
5,null,null,Time to think about what I used to have and what I no longer do have
6,null,null,I think  if I could only be with her again
25 2
 (0, 0),
1,null,null,It used to be talking about nothing for hours  now it is just not talking
2,null,null,Really  what happened
26 1
 (0, 0),
1,null,null,I can tell by the end of the night I am gon na cut my arm
```

Figure 57: A sample of the Twitter dataset for the ECPE task

ecpe_raw_long_summarized_max_5_clauses.txt

```
71 2
 (0, 0),
1,null,null,I am a 42 year old woman who is single and has no one  special  in my life
2,null,null,I really want to make 2006 my year
72 4
 (0, 0),
1,null,null,I am glad you joined
2,null,null,This board is a great place for information and support
3,null,null,I think you did great starting on both talk therapy and medicines
4,null,null,Tom  thanks for joining
74 2
 (0, 0),
1,null,null,I am so depressed I do not know what to do
2,null,null,Dropped out of school  have been unemployed for two months  have absolutely no friends and spend my days browsing reddit
76 3
 (0, 0),
1,null,null,Girlfriend attempted to break her arm to avoid going to work
2,null,null,My girlfriend began her summer job in a hospital earlier this summer where she mostly cleans and prepares meals for the patients
3,null,null,She unwillingly explained that she had tried to break her arm in order to not have to go to work anymore for the rest of the summer
82 4
 (0, 0),
1,null,null,How are you supposed to feel when you take them  amp  they are effective
2,null,null,i have been feeling depressed since i was 19  i am 21 now
3,null,null,i loved how positive i was feeling
4,null,null,is this how antidepressants are supposed to make you feel if they re working
```

Figure 58: A sample of the summarised Reddit dataset for the ECPE task

After pre-processing of the text data, the Twitter and Reddit datasets are manually annotated with the emotion and the cause for each entry. Instead of six emotions in the NTCIR-13 dataset, there is only one emotion in the annotated datasets, which is ‘depressed’. Figure 59 shows a sample of an annotated text entry.

```
77 6
 (6, 4),
1,null,null,My Anxiety Disorder  PTSD  Depression are real and
2,null,null,today I am battling
3,null,null,My happiness is fading and
4,null,null,my brain is messing with me
5,null,null,so badly  I have been trying hard not to cry  trying hard just to stay positive
6,depressed,struggling,but I am struggling
```

Figure 59: An annotated text entry sample

From Figure 59, the explanation of the data format is as follows:

- 77 6 - The data has the index 77 and this piece of data contains 6 clauses
- (6, 4) - This data's emotion is in clause 6 and the cause is in clause 4
- null - This clause does not contain any emotion or cause clause
- depressed, struggling - 'depressed' is the clause with the emotion and 'struggling' is the secondary emotion that can be found in the particular clause

After annotating the two datasets, several models were trained with different hyperparameters to see which set of hyperparameters produce the best performing F1-score for the pair extraction.

9.4 Hyperparameters

These are the hyperparameters held constant for both variants of ECPE:

- Embedding Dimension (word) - 200
- Embedding Dimension (position) - 30
- Max Sentence Length - 30
- Max Document Length - 41
- L2 Regularisation - 1e-04
- Epochs - 15 (Twitter dataset) or 20 (Reddit dataset)

These are the hyperparameters being experimented for both variants of ECPE:

- Number of Hidden Unit (n_hid)
- Batch Size (BS)
- Learning Rate (LR)
- Diminish Factor (dim_f)

9.5 Results

The tables below show the performances of all the models trained with different hyperparameters, different datasets and different variants of ECPE model, and also to highlight the best performing one in each category. Similarly, to performing ECPE on the benchmark NTCIR-13 dataset, precision (P), recall (R) and F1-score (F1) are used as metrics and F1-score of the pair extraction is the main metric to compare the models' performances.

9.5.1 E2E-PExt(E) on Twitter Dataset

Hyperparameters				Emotion Extraction			Cause Extraction			Pair Extraction		
n_hid	**BS**	**LR**	**dim_f**	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
128	**64**	**0.005**	**0.400**	0.6706	0.6333	0.6514	0.7778	0.4667	0.5833	0.5584	0.4778	**0.5150**
128	64	0.005	0.200	0.7941	0.6000	0.6835	0.7541	0.5111	0.6093	0.4731	0.4889	0.4809
128	64	0.001	0.400	0.6310	0.5889	0.6092	0.6986	0.5667	0.6258	0.6250	0.3333	0.4348
128	64	0.001	0.200	0.6986	0.5667	0.6258	0.6622	0.5444	0.5976	0.4444	0.5333	0.4848
128	128	0.005	0.400	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
128	128	0.005	0.200	0.8571	0.2667	0.4068	0.8250	0.3667	0.5077	0.4103	0.3556	0.3810
128	128	0.001	0.400	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
128	128	0.001	0.200	0.6818	0.3333	0.4478	0.6727	0.4111	0.5103	0.5385	0.3111	0.3944
256	64	0.005	0.400	0.6429	0.7000	0.6702	0.6986	0.5667	0.6258	0.5526	0.4667	0.5060
256	64	0.005	0.200	0.7568	0.6222	0.6829	0.6714	0.5222	0.5875	0.5000	0.5000	0.5000
256	64	0.001	0.400	0.7612	0.5667	0.6497	0.7586	0.4889	0.5946	0.5634	0.4444	0.4969
256	64	0.001	0.200	0.7237	0.6111	0.6627	0.6957	0.5333	0.6038	0.4141	0.5889	0.4862
256	128	0.005	0.400	0.6667	0.4000	0.5000	0.6667	0.5333	0.5926	0.5082	0.3444	0.4106
256	128	0.005	0.200	0.6027	0.4889	0.5399	0.5700	0.6333	0.6000	0.3101	0.5444	0.3952
256	128	0.001	0.400	0.5595	0.5222	0.5402	0.5417	0.5778	0.5591	0.4286	0.3333	0.3750
256	128	0.001	0.200	0.6269	0.4667	0.5350	0.6575	0.5333	0.5890	0.3672	0.5222	0.4312

Table 32: Performances of the E2E-PExt(E) model on the Twitter Dataset for various hyperparameters

9.5.2 E2E-PExt(E) on Reddit Dataset

Hyperparameters				Emotion Extraction			Cause Extraction			Pair Extraction		
n_hid	**BS**	**LR**	**dim_f**	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
128	64	0.005	0.400	0.8182	0.6818	0.7438	0.6667	0.4242	0.5185	0.5410	0.5000	0.5197
128	64	0.005	0.200	0.7460	0.7121	0.7287	0.5937	0.5758	0.5846	0.4483	0.5909	0.5098
128	64	0.001	0.400	0.7000	0.6364	0.6667	0.6066	0.5606	0.5827	0.5000	0.4545	0.4762
128	64	0.001	0.200	0.6716	0.6818	0.6767	0.5833	0.5303	0.5556	0.3333	0.6364	0.4375
128	128	0.005	0.400	0.7333	0.6667	0.6984	0.6296	0.5152	0.5667	0.4474	0.5152	0.4789
128	128	0.005	0.200	0.7857	0.6667	0.7213	0.6000	0.5000	0.5455	0.3796	0.6212	0.4713
128	128	0.001	0.400	0.7027	0.3939	0.5049	0.6122	0.4545	0.5217	0.8667	0.1970	0.3210
128	128	0.001	0.200	0.5870	0.4091	0.4821	0.6250	0.3030	0.4082	0.4600	0.3485	0.3966
256	64	0.005	0.400	0.6500	0.5909	0.6190	0.6000	0.5909	0.5954	0.5833	0.4242	0.4912
256	64	0.005	0.200	0.7358	0.5909	0.6555	0.6875	0.5000	0.5789	0.4444	0.5455	0.4898
256	**64**	**0.001**	**0.400**	0.7742	0.7273	0.7500	0.5541	0.6212	0.5857	0.4592	0.6818	**0.5488**
256	64	0.001	0.200	0.7727	0.5152	0.6182	0.6800	0.5152	0.5862	0.4375	0.5303	0.4795
256	128	0.005	0.400	0.7407	0.6061	0.6667	0.5932	0.5303	0.5600	0.5357	0.4545	0.4918
256	128	0.005	0.200	0.6316	0.1818	0.2824	0.8333	0.3030	0.4444	0.4464	0.3788	0.4098
256	128	0.001	0.400	0.8400	0.6364	0.7241	0.5902	0.5455	0.5669	0.5000	0.4848	0.4923
256	128	0.001	0.200	0.7736	0.6212	0.6891	0.6226	0.5000	0.5546	0.3933	0.5303	0.4516

Table 33: Performances of the E2E-PExt(E) model on the Reddit Dataset for various hyperparameters

9.5.3 E2E-PExt(C) on Twitter Dataset

Hyperparameters				Emotion Extraction			Cause Extraction			Pair Extraction		
n_hid	**BS**	**LR**	**dim_f**	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
128	64	0.005	0.400	0.7500	0.6000	0.6667	0.7164	0.5333	0.6115	0.5181	0.4778	0.4971
128	64	0.005	0.200	0.8065	0.5556	0.6579	0.7368	0.4667	0.5714	0.5065	0.4333	0.4671
128	64	0.001	0.400	0.6543	0.5889	0.6199	0.7222	0.57781	0.6420	0.5278	0.4222	0.4691
128	64	0.001	0.200	0.6322	0.6111	0.6215	0.7353	0.5556	0.6329	0.4153	0.5444	0.4712
128	128	0.005	0.400	0.5349	0.5111	0.5227	0.5876	0.6333	0.6096	0.8750	0.0778	0.1429
128	128	0.005	0.200	0.7317	0.3333	0.4580	0.7959	0.4333	0.5612	0.4023	0.3889	0.3955
128	128	0.001	0.400	0.3309	1.0000	0.4972	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
128	128	0.001	0.200	0.6739	0.3444	0.4559	0.7045	0.3444	0.4627	0.6667	0.1333	0.2222
256	64	0.005	0.400	0.7097	0.4889	0.5789	0.6714	0.5222	0.5875	0.5000	0.3889	0.4375
256	64	0.005	0.200	0.6479	0.5111	0.5714	0.7231	0.5222	0.6065	0.3788	0.5556	0.4505
256	**64**	**0.001**	**0.400**	0.7746	0.6111	0.6832	0.6866	0.5111	0.5860	0.5443	0.4778	**0.5089**
256	64	0.001	0.200	0.7125	0.6333	0.6706	0.6849	0.5556	0.6135	0.4608	0.5222	0.4896
256	128	0.005	0.400	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
256	128	0.005	0.200	0.0000	0.0000	0.0000	1.0000	0.0111	0.0220	0.3483	0.3444	0.3464
256	128	0.001	0.400	0.5408	0.5889	0.5638	0.5833	0.6222	0.6022	0.4928	0.3778	0.4277
256	128	0.001	0.200	0.7018	0.4444	0.5442	0.6818	0.5000	0.5769	0.3121	0.4889	0.3810

Table 34: Performances of the E2E-PExt(C) model on the Twitter Dataset for various hyperparameters

9.5.4 E2E-PExt(C) on Reddit Dataset

Hyperparameters				Emotion Extraction			Cause Extraction			Pair Extraction		
n_hid	**BS**	**LR**	**dim_f**	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
128	64	0.005	0.400	0.7719	0.6667	0.7154	0.5714	0.5455	0.5581	0.5070	0.5455	0.5255
128	64	0.005	0.200	0.7377	0.6818	0.7087	0.5968	0.5606	0.5781	0.4270	0.5758	0.4903
128	64	0.001	0.400	0.7541	0.6970	0.7244	0.6212	0.6212	0.6212	0.4684	0.5606	0.5103
128	64	0.001	0.200	0.6719	0.6515	0.6615	0.5882	0.6061	0.5970	0.4037	0.6667	0.5029
128	128	0.005	0.400	0.7377	0.6818	0.7087	0.6182	0.5152	0.5620	0.4583	0.5000	0.4783
128	128	0.005	0.200	0.7656	0.7424	0.7538	0.6250	0.6061	0.6154	0.3894	0.6667	0.4916
128	128	0.001	0.400	0.6905	0.4394	0.5370	0.6364	0.5303	0.5785	0.9091	0.1515	0.2597
128	128	0.001	0.200	0.7692	0.3030	0.4348	0.5741	0.4697	0.5167	0.3750	0.4545	0.4110
256	64	0.005	0.400	0.7049	0.6515	0.6772	0.6379	0.5606	0.5968	0.5075	0.5152	0.5113
256	64	0.005	0.200	0.6875	0.6667	0.6769	0.6379	0.5606	0.5968	0.4430	0.5303	0.4828
256	**64**	**0.001**	**0.400**	0.7302	0.6970	0.7132	0.6129	0.5758	0.5937	0.4762	0.6061	**0.5333**
256	64	0.001	0.200	0.7800	0.5909	0.6724	0.6207	0.5455	0.5806	0.4507	0.4848	0.4672
256	128	0.005	0.400	0.5818	0.4848	0.5289	0.5538	0.5455	0.5496	0.5333	0.3636	0.4324
256	128	0.005	0.200	0.7273	0.6061	0.6612	0.6591	0.4394	0.5273	0.3980	0.5909	0.4756
256	128	0.001	0.400	0.7667	0.6970	0.7302	0.6167	0.5606	0.5873	0.5660	0.4545	0.5042
256	128	0.001	0.200	0.7679	0.6515	0.7049	0.5965	0.5152	0.5528	0.3565	0.6212	0.4530

Table 35: Performances of the E2E-PExt(C) model on the Reddit Dataset for various hyperparameters

9.5.5 Best Performing Models

Table 36 shows the best performing models in terms of Pair Extraction F1-score for the two datasets and the two variants of ECPE models.

Variant	Dataset	Hyperparameters				Pair Extraction		
		n_hid	BS	LR	dim_f	P	R	F1
E2E-PExt(E)	Twitter	128	64	0.005	0.400	0.5584	0.4778	**0.5150**
	Reddit	256	64	0.001	0.400	0.4592	0.6818	**0.5488**
E2E-PExt(C)	Twitter	256	64	0.001	0.400	0.5443	0.4778	**0.5089**
	Reddit	256	64	0.001	0.400	0.4762	0.6061	**0.5333**

Table 36: Best performing models in terms of Pair Extraction F1 Score

From Table 36, the Pair Extraction F1 score is relatively close to the model trained on the benchmark dataset. Also, the model variant E2E-PExt(E) performs slightly better than the variant E2E-PExt(C) on both of the data.

9.6 Limitations

The datasets used in this task were labelled by an undergraduate student from SCSE. Also, annotating the causes of depression might not be as straightforward as annotating the text as ‘depressive’ or ‘not depressive’ because of the way the text is being written, the domain knowledge of depression and also the understanding of English. As such, there might be a bias in annotation which may make the Pair Extraction F1-score higher or lower than the values presented. There is a need for experts like professionals in the field of Psychology or Linguistics to annotate the data so that the annotations are unbiased and prediction of F1 scores are more accurate, but due to the lack of time and cost, this has to be compromised.

9.7 Conclusion for this chapter on Emotion-Cause Pair Extraction

In this chapter, the ECPE task is being thoroughly explained, including previous research and the two variants of models proposed. ECPE is then applied into the context of depression and see if the model can also extract the emotion-cause pair from the manually annotated depressive Twitter and Reddit dataset. The limitations of this chapter are also discussed due to the potential bias in the annotation of the datasets.

CHAPTER 10: CONCLUSION AND FUTURE WORK

10.1 Conclusion

In this project, we looked at the three main tasks of Emotion Classification, Emotion Intensity Prediction and Emotion-cause Pair Extraction, and one subtask on Text summarisation. We also did some analysis on the Toy dataset, Twitter dataset and Reddit dataset.

In the Emotion Classification task, we looked at the sequence models and the transformer models to predict whether a text is depressive or not by looking at the various metrics. Then, we observed that the model trained with the Toy dataset is too good to be true and hence, we did some manual observations and inter-annotator agreements to conclude that the Toy Dataset is deemed unsuitable to be used in further tasks.

In the Emotion Intensity Prediction task, we looked at two methods to obtain depression scores for the entries in the Twitter and Reddit dataset. One way is to adopt the combination of symbolic and subsymbolic approach, where the composite score from VADER, text2emotion, and the training of models from the three different WASSA 2017 EmoInt emotion datasets. The other more straightforward way is to leverage on the use of the SenticNet API on Depression Identification. Then, we made a comparison between the two methods and concluded that there does not exist the best way to obtain the depression score as there are no golden annotations available by experts from the field of Psychology or Linguistics.

We then moved on to discuss text summarisation and the difference between Extractive and Abstractive text summarisation. Then, we leverage on the use of Extractive text summarisation to summarise the long Reddit dataset for the next main task which is the Emotion-Cause Pair Extraction (ECPE).

Lastly, we discussed ECPE by looking at what it does from a research paper and its performances. Then, we describe the two variants of ECPE and how these variants can model the manually labelled depressive Twitter and Reddit datasets. Next, we experimented with various hyperparameters to see which model had the best performance based on the different datasets and also the model variant. Lastly, we concluded that there might be a bias in the annotation of the data due to the lack of experts from the field of Psychology or Linguistics.

10.2 Future Work

The performance of the tasks stated seems promising. Hence, a future work can be to focus on model deployment on a website or a phone application, where the users can input a piece of text that is retrieved from social media sites, then the application will return whether that piece of text contains depressive contents, the intensity of depression and the cause of depression.

Additionally, a ‘report’ button can be created. In the event where a depressive text has a very high depression intensity score or the predicted cause of depression is lethal, the users can click on the button. Relevant authorities will soon be notified and take immediate and the correct actions to curb one’s depression.

REFERENCES

1. Cooper, J. and Bhandari, S., 2021. *The Link Between Stress and Depression*. [online] WebMD. Available at: <https://www.webmd.com/depression/features/stress-depression> [Accessed 4 October 2021].
2. Legg, T. and Higuera, V., 2021. *Everything You Want to Know About Depression*. [online] Healthline. Available at: <https://www.healthline.com/health/depression> [Accessed 4 October 2021].
3. M. De Choudhury, M. Gammon, S. Counts, and E. Horvitz, "Predicting Depression via Social Media," ICWSM, vol. 7, no. 1, Jun. 2013 [Accessed 4 October 2021].
4. Q. Hu, A. Li, F. Heng, J. Li, and T. Zhu, "Predicting Depression of Social Media User on Different Observation Windows," presented at the 2015 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT), 2015 [Accessed 4 October 2021].
5. Anwar, S., 2016. *Sentiment Analysis versus Emotional Analysis: Same or Different?*. [online] Linkedin.com. Available at: <https://www.linkedin.com/pulse/sentiment-analysis-versus-emotional-same-different-shahbaz-anwar/> [Accessed 4 October 2021].
6. Singh, D. and Wang, A., 2015. Detecting Depression Through Tweets. *Stanford*, [online] Available at: <https://web.stanford.edu/class/archive/cs/cs224n/cs224n.1184/reports/6879557.pdf> [Accessed 4 October 2021].
7. Medium. 2021. *Targeted Sentiment analysis vs Traditional Sentiment analysis*. [online] Available at: <https://towardsdatascience.com/targeted-sentiment-analysis-vs-traditional-sentiment-analysis-4d9f2c12a476> [Accessed 4 October 2021].
8. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., N. Gomez, A., Kaiser, L. and Polosukhin, I., 2017. Attention Is All You Need. *arXiv*, [online] (5). Available at: <https://arxiv.org/pdf/1706.03762v5.pdf> [Accessed 4 October 2021].
9. Medium. 2019. *What is a Transformer?*. [online] Available at: <https://medium.com/inside-machine-learning/what-is-a-transformer-d07dd1fbec04> [Accessed 4 October 2021].

10. Affane, R., 2020. *Understanding the Hype Around Transformer NLP Models*. [online] Blog.dataiku.com. Available at: <https://blog.dataiku.com/decoding-nlp-attention-mechanisms-to-understand-transformer-models> [Accessed 4 October 2021].
11. Kaur, J., 2019. *Evolution and Future of Natural Language Processing (NLP)*. [online] Xenonstack.com. Available at: <https://www.xenonstack.com/blog/evolution-of-nlp> [Accessed 4 October 2021].
12. Cavaliere, Danilo, Senatore, Sabrina and Loia, Vincenzo, "Data-Information-Concept Continuum From a Text Mining Perspective," in Encyclopedia of Bioinformatics and Computational Biology, Ranganathan, Shoba, Gribskov, Michael, Nakai, Kenta and Schönbach, Christian, Eds., Oxford, Academic Press, 2019, pp. 586-601.
13. Thanaki, J., 2021. *Python Natural Language Processing*. [online] O'Reilly Online Learning. Available at: <https://www.oreilly.com/library/view/python-natural-language/9781787121423/831c05a6-edae-4475-91f9-8550c3b7f8ea.xhtml> [Accessed 4 October 2021].
14. Liddy, Elizabeth D, "Natural language processing," in Encyclopedia of Library and Information Science, 2 ed., Marcel Decker, Ed., 2001.
15. Brownlee, J., 2021. *What is Deep Learning?*. [online] Machine Learning Mastery. Available at: <https://machinelearningmastery.com/what-is-deep-learning/> [Accessed 4 October 2021].
16. Education, I., 2021. *What are Recurrent Neural Networks?*. [online] Ibm.com. Available at: <https://www.ibm.com/cloud/learn/recurrent-neural-networks> [Accessed 10 October 2021].
17. Amidi, A. and Amidi, S., 2021. *CS 230 - Recurrent Neural Networks Cheatsheet*. [online] Stanford.edu. Available at: <https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-recurrent-neural-networks> [Accessed 10 October 2021].
18. Feng, C., 2021. *Vanilla Recurrent Neural Network - Machine Learning Notebook*. [online] Calvinfeng.gitbook.io. Available at: <https://calvinfeng.gitbook.io/machine-learning-notebook/supervised-learning/recurrent-neural-network/recurrent_neural_networks> [Accessed 10 October 2021].

19. Phi, M., 2021. *Illustrated Guide to LSTM's and GRU's: A step by step explanation*. [online] Medium. Available at: <https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21> [Accessed 10 October 2021].
20. Maxime, M., 2021. *What is a Transformer?*. [online] Medium. Available at: <https://medium.com/inside-machine-learning/what-is-a-transformer-d07dd1fbec04> [Accessed 10 October 2021].
21. Devlin, J., Chang, M., Lee, K. and Toutanova, K., 2021. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. [online] p.16. Available at: <https://arxiv.org/pdf/1810.04805.pdf> [Accessed 10 October 2021].
22. Karani, D., 2021. *Introduction to Word Embedding and Word2Vec*. [online] Medium. Available at: <https://towardsdatascience.com/introduction-to-word-embedding-and-word2vec-652d0c2060fa> [Accessed 10 October 2021].
23. En.wikipedia.org. 2021. *Word embedding - Wikipedia*. [online] Available at: <https://en.wikipedia.org/wiki/Word_embedding> [Accessed 10 October 2021].
24. Zulkifli, H., 2021. *Understanding Learning Rates and How It Improves Performance in Deep Learning*. [online] Medium. Available at: <https://towardsdatascience.com/understanding-learning-rates-and-how-it-improves-performance-in-deep-learning-d0d4059c1c10> [Accessed 10 October 2021].
25. Gaillard, F., 2021. *Batch size (machine learning) | Radiology Reference Article | Radiopaedia.org*. [online] Radiopaedia.org. Available at: <https://radiopaedia.org/articles/batch-size-machine-learning> [Accessed 10 October 2021].
26. Brownlee, J., 2021. *Difference Between a Batch and an Epoch in a Neural Network*. [online] Machine Learning Mastery. Available at: <https://machinelearningmastery.com/difference-between-a-batch-and-an-epoch/> [Accessed 10 October 2021].
27. Doshi, S., 2021. *Various Optimization Algorithms For Training Neural Network*. [online] Medium. Available at: <https://towardsdatascience.com/optimizers-for-training-neural-network-59450d71caf6> [Accessed 10 October 2021].

28. Riva, M., 2021. *Interpretation of Loss and Accuracy for a Machine Learning Model*. [online] baeldung. Available at: <https://www.baeldung.com/cs/ml-loss-accuracy> [Accessed 10 October 2021].
29. Wood, T., 2021. *Precision and Recall*. [online] DeepAI. Available at: <https://deepai.org/machine-learning-glossary-and-terms/precision-and-recall> [Accessed 10 October 2021].
30. Data School. 2021. *Simple guide to confusion matrix terminology*. [online] Available at: <https://www.dataschool.io/simple-guide-to-confusion-matrix-terminology/> [Accessed 10 October 2021].
31. Narkhede, S., 2021. *Understanding AUC - ROC Curve*. [online] Medium. Available at: <https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5> [Accessed 10 October 2021].
32. En.wikipedia.org. 2021. *Receiver operating characteristic - Wikipedia*. [online] Available at: <https://en.wikipedia.org/wiki/Receiver_operating_characteristic> [Accessed 10 October 2021].
33. Corpus Linguistic Methods. 2021. *What is inter-annotator agreement?*. [online] Available at: <https://corpuslinguisticmethods.wordpress.com/2014/01/15/what-is-inter-annotator-agreement/> [Accessed 10 October 2021].
34. Pykes, K., 2021. *Cohen's Kappa*. [online] Medium. Available at: <https://towardsdatascience.com/cohens-kappa-9786ceceab58> [Accessed 10 October 2021].
35. Wu, S., 2021. *What are the best metrics to evaluate your regression model?*. [online] Medium. Available at: <https://towardsdatascience.com/what-are-the-best-metrics-to-evaluate-your-regression-model-418ca481755b> [Accessed 10 October 2021].
36. Romero, V., 2021. *GitHub - viritaromero/Detecting-Depression-in-Tweets: Detecting Depression in Tweets using Baye's Theorem*. [online] GitHub. Available at: <https://github.com/viritaromero/Detecting-Depression-in-Tweets> [Accessed 10 October 2021].
37. Wang, S., 2021. *GitHub - swcwang/depression-detection: Depression Detection Using Twitter Data - Group project for Udacity Private & Secure AI Project Showcase*. [online]

GitHub. Available at: <https://github.com/swcwang/depression-detection> [Accessed 10 October 2021].

38. Pirina, I., 2021. *Identifying-depression/Data_Collector at master · Inusette/Identifying-depression*. [online] GitHub. Available at: <https://github.com/Inusette/Identifying-depression/tree/master/Data_Collector> [Accessed 10 October 2021].
39. Bravom, F., 2021. *CodaLab - Competition*. [online] Competitions.codalab.org. Available at: <https://competitions.codalab.org/competitions/16380> [Accessed 10 October 2021].
40. E Cambria et al. SenticNet 6: Ensemble Application of Symbolic and Subsymbolic AI for Sentiment Analysis. In: Proceedings of CIKM, 105-114 (2020) — sentic.net/senticnet-6.pdf
41. Cambria, E., 2021. *Sentic Computing « SenticNet*. [online] Sentic.net. Available at: <https://sentic.net/computing/> [Accessed 10 October 2021].
42. Cambria, E., 2021. *About « SenticNet*. [online] Sentic.net. Available at: <https://sentic.net/about/> [Accessed 10 October 2021].
43. DeepAI. 2021. *Text Summarization*. [online] Available at: <https://deepai.org/machine-learning-model/summarization> [Accessed 10 October 2021].
44. Hutto, C. and Gilbert, E., 2014. *VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text*. [online] Aaai.org. Available at: <https://www.aaai.org/ocs/index.php/ICWSM/ICWSM14/paper/viewFile/8109/8122> [Accessed 10 October 2021].
45. Band, A., 2020. *Text2emotion: Python package to detect emotions from textual data*. [online] Medium. Available at: <https://towardsdatascience.com/text2emotion-python-package-to-detect-emotions-from-textual-data-b2e7b7ce1153> [Accessed 10 October 2021].
46. Singh, A., 2019. *Building State-of-the-Art Language Models with BERT*. [online] Medium. Available at: <https://medium.com/saarthi-ai/bert-how-to-build-state-of-the-art-language-models-59dddfa9ac5d> [Accessed 10 October 2021].
47. Ahmed, S., 2016. *Plutchik's Wheel of Emotions*. [online] ResearchGate. Available at: <https://www.researchgate.net/figure/Plutchiks-Wheel-of-Emotions-as-cited-in-20_fig6_301016353> [Accessed 10 October 2021].

48. Betterhealth.vic.gov.au. 2021. *Depression explained - Better Health Channel*. [online] Available at: <https://www.betterhealth.vic.gov.au/health/conditionsandtreatments/depression> [Accessed 10 October 2021].
49. Cheung, J., 2008. *Comparing Abstractive and Extractive Summarization of Evaluative Text*. [online] Cs.toronto.edu. Available at: <http://www.cs.toronto.edu/~jcheung/papers/honours-thesis.pdf> [Accessed 10 October 2021].
50. Weng, J., 2021. *How to Perform Abstractive Summarization with PEGASUS*. [online] Medium. Available at: <https://towardsdatascience.com/how-to-perform-abstractive-summarization-with-pegasus-3dd74e48bafb> [Accessed 10 October 2021].
51. Zhang, J., Zhao, Y., Saleh, M. and Liu, P., 2021. *PEGASUS: Pre-training with Extracted Gap-sentences for Abstractive Summarization*. [online] arXiv.org. Available at: <https://arxiv.org/abs/1912.08777> [Accessed 10 October 2021].
52. Huggingface.co. 2021. *Models - Hugging Face*. [online] Available at: <https://huggingface.co/models?search=pegasus> [Accessed 10 October 2021].
53. Singh, A., Hingane, S., Wani, S. and Modi, A., 2021. *An End-to-End Network for Emotion-Cause Pair Extraction*. [online] arXiv.org. Available at: <https://arxiv.org/abs/2103.01544> [Accessed 10 October 2021].
54. Research.nii.ac.jp. 2021. *NTCIR-13 / NTCIR*. [online] Available at: <http://research.nii.ac.jp/ntcir/ntcir-13/> [Accessed 10 October 2021].