Department of Computer Science
University of Hong Kong

Interim Report

Deep Learning Based Public Sentiment Analysis System on Trending News

Supervisor:
Dr. Kong Lingpeng

Students:
Woo Nathan Hugo 3035683162
Yeung Sum 3035683150

Date of Submission: 22/1/2023
Abstract

In the era of the Internet, social media is a common place for the public to express their attitudes on various issues. Different government organizations also announce their newest policies on social media. Through surfing the Internet and reading social media, one can know what the public thinks about popular topics. However, it is very time-consuming to gather and analyze the opinions of the public. Hence, the main objective of this project is to build an automatic public sentiment analysis system on general trending news. Currently, data have been collected from Twitter and Reddit for model training and used for training. However, the model does not perform as we expected. In the next phase of the project, improvements will be made to the current model and we will also design and build a webpage for displaying the result of this project.
Acknowledgement

We would like to express our gratitude to my supervisor, Dr. Kong Lingpeng. Dr. Kong gave us valuable insights and guidance on doing the project. He also gave various suggestions for us to consider when we were facing difficulties. We sincerely thank Dr. Kong for his support and encouragement.
# Table of Contents

Abstract ................................................................................................................................. 2

Acknowledgement .................................................................................................................. 3

List of Figures ......................................................................................................................... 5

List of Tables .......................................................................................................................... 6

Abbreviation .......................................................................................................................... 7

1. Introduction ....................................................................................................................... 8

1.1 Background ...................................................................................................................... 8

1.2 Objective ......................................................................................................................... 8

1.3 Scope and Deliverables .................................................................................................... 8

1.4 Benefits ............................................................................................................................ 9

1.5 Outline of the report ....................................................................................................... 9

2. Methodology .................................................................................................................... 10

2.1 Data Collection and Usage ............................................................................................. 10

2.2 Model Architecture ........................................................................................................ 11

2.2.1 Autoencoder and Variational AutoEncoder (VAE) .................................................... 11

2.2.2 Simple Contrastive Learning of Sentence Embeddings (SimCSE) ......................... 12

2.2.3 Discussion .................................................................................................................. 12

2.3 Evaluation ....................................................................................................................... 13

2.4 User Interface for Result Delivery ................................................................................ 14

3. Current progress and preliminary result ......................................................................... 15

3.1 Data Collection .............................................................................................................. 15

3.2 Work Done for Model Architecture ............................................................................... 15

3.3 Preliminary Result for the Model Training .................................................................... 16

3.3.1 Vanishing KL Divergence Loss ................................................................................ 16

3.3.2 Visualization of the Latent Vectors ......................................................................... 16

3.4 Limitations ...................................................................................................................... 19

4. Future Works ................................................................................................................... 20

4.1 Project Schedule ............................................................................................................ 20

4.2 Possible Improvements .................................................................................................. 20

5. Conclusion ....................................................................................................................... 21

References ............................................................................................................................ 22

Appendices ............................................................................................................................ 24
List of Figures

Figure 2.1 Architecture of an autoencoder .........................................................p.11
Figure 3.1 Box plots of the cosine distance of paired text for different RNNs ..........p.16
Figure 3.2 Box plots of the cosine distance of paired text for supervised and
    unsupervised SimCSE using pretrained BERT ..............................................p.17
Figure 3.3 Latent codes distributions after UMAP of the two models .....................p.18
Figure 3.4 The search results of similar embeddings with the input sentence from
    different models .......................................................................................p.19
List of Tables

Table 4.1 Renewed Project Schedule ..............................................................................p.20
### Abbreviation

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>API</td>
<td>Application Programming Interface</td>
</tr>
<tr>
<td>BERT</td>
<td>Bidirectional Encoder Representations from Transformers</td>
</tr>
<tr>
<td>BLEU</td>
<td>Bilingual evaluation understudy</td>
</tr>
<tr>
<td>CNN</td>
<td>Convolutional Neural Network</td>
</tr>
<tr>
<td>KL</td>
<td>Kullback–Leibler</td>
</tr>
<tr>
<td>LSTM</td>
<td>Long Short-Term Memory</td>
</tr>
<tr>
<td>NLP</td>
<td>Natural Language Processing</td>
</tr>
<tr>
<td>RNN</td>
<td>Recurrent Neural Network</td>
</tr>
<tr>
<td>SimCSE</td>
<td>Simple Contrastive Learning of Sentence Embeddings</td>
</tr>
<tr>
<td>SNLI</td>
<td>Stanford Natural Language Inference</td>
</tr>
<tr>
<td>UMAP</td>
<td>Uniform Manifold Approximation and Projection</td>
</tr>
<tr>
<td>VAE</td>
<td>Variational Autoencoder</td>
</tr>
</tbody>
</table>
1. Introduction

Section 1.1 explains the background of the topic and the drawbacks of the current system. Section 1.2 describes the objectives of the project. Section 1.3 talks about the scope and deliverables of the project. Section 1.4 discusses the benefits of the project. Section 1.5 showcases the outline of the report.

1.1 Background

Through sentiment analysis, one can gather thoughts and subjectivity from people toward others or topics (Medhat et al., 2014). Sentiment analysis has been used in multiple fields as it can acknowledge and predict others’ actions. In recent years, more and more people are using social media to share their life and express their opinion on different topics. Through browsing social media, one can know society’s attitudes on various social concerns, such as whether the public is supporting or is against certain issues. Despite the fact that some public sentiment analysis projects are using Twitter data, most of the projects focus on specific topics and issues, such as COVID-19 or the presidential election (Kausar et al., 2021; Wang et al., 2012). The research and analysis involved in these projects take a considerable amount of time to complete, and one will only be able to obtain the public's opinion on certain topics once the analyses are completed and the paper is published. Apart from the researchers, individuals or groups of people such as companies or the government probably want to know the public attitudes and opinions toward certain issues. However, it is very time-consuming to gather and analyze the opinions of a large group of people due to the huge information size, or using some traditional methods like questionnaires.

1.2 Objective

The purpose of this project is to conduct sentiment analysis on more general popular topics. The most common opinions among a large group of people who are using the social media platform will be extracted, their attitudes will be shown by a sentiment score along with the opinion. With the help of the latest technologies, it is expected that the system would be performing well in accomplishing the summary and analysis tasks.

1.3 Scope and Deliverables

The opinions of the public are collected from Twitter and Reddit while the general popular news is mainly from Western countries like America or the UK, and they are collected
from Google Trends. To deal with the difficult sentiment analysis tasks which involve Natural Language Processing (NLP), various machine learning and deep learning methods will be used. Therefore, in this project, we are going to build an automated system that potentially uses many novel technologies, such as Variational Autoencoder, Transformers and pre-trained models for analyzing public sentiments regarding the trending topics in the community. Since this project is mainly built for individuals, instead of having the analysis on certain input-targeted topics, the automated system will automatically pick up some trending news happening in the community for analysis. Since the whole system does not require human intervention after all the components are built, it will be fast if the computation power is high enough. A website will be also deployed to show the result of the system after the system is ready to use. The current progress is on track, and the opinions and data have been collected and cleaned for future model training.

1.4 Benefits

Users can know the public attitudes and opinions towards trending news easily through the website, which saves a lot of time and human power for conducting research or looking through a large number of comments and posts on social media platforms. For the controversial news, since both sides of the argument would probably be shown at the same time, the system can also help individuals in understanding what the people on the opposing side are thinking. Moreover, the website for displaying the result will be updated on a daily or weekly basis, which allows one to understand the public’s thoughts on the daily or weekly popular issues. Furthermore, by utilizing this automatic public sentiment analysis system, researchers can conduct research that requires an understanding of public sentiment, or at least gain some insights, without having to conduct a questionnaire. Additionally, since the process of data acquisition and sentiment analysis for user-inputted topics and this project are almost the same, this project can be further extended for the use of other groups of interest.

1.5 Outline of the report

This report consists of multiple sections. Section 2 discusses the methodology involved in this project. Section 3 showcases the current progress and preliminary results of the whole project. Section 4 mentions the work to be done in the future. Section 5 gives a brief conclusion of the report.
2. Methodology

The system collects data from multiple application programming interfaces (API) and uses deep learning and machine learning models to summarize and analyze the data. There will also be a webpage for displaying the result. The content on the webpage will be updated on a daily or weekly basis.

Section 2.1 first talks about how the data are collected and how they will be used in this project. Section 2.2 explains the proposed method to generate the sentiment analysis by the system. Section 2.3 discusses how to evaluate the performance of the model. Section 2.4 talks about the webpage for displaying the result.

2.1 Data Collection and Usage

This project gathers opinions of the general public from Twitter and Reddit because they are two of the biggest and most popular social media nowadays. The system uses the Twitter API and Reddit API to collect the data from Twitter and Reddit. A popular third-party API for Google Trends is also used to collect the data for trending news since it can show the popular searches in recent hours, which represents some popular topics. The data is obtained legally by the APIs and is only be used for this project. The system first uses Google Trends to get the most popular and recently searched keywords. Using the keywords, the system searches and gets comments that are related to popular topics on social media platforms, which are Twitter and Reddit.

Due to the different content presentation designs on Twitter and Reddit, the news is spread at a different speed and has a different lifespan on the two platforms (Priya et al., 2019). Reddit is suitable for getting information at the early phase of the event and Twitter is suitable for discussing the news and having updates for a longer period since it is easy for the author to extend the Tweets by quoting the previous Tweet. Therefore, the information gained from different platforms is different in some way and potentially has some sort of bias toward the issue. In hope of getting a more representative result, multiple sources of comment data will be used, more platforms can be used for future development.

After obtaining the data, the data are pre-processed and cleaned, such as filtering links or hashtags to increase the effectiveness of training the model (Sahayak et al., 2015). For training, the data will be split into different ratios of the training, development and testing
dataset; and will be evaluated on different models. Their performance will be compared, and the best ratio will be chosen.

2.2 Model Architecture

The system will use deep learning and NLP models to convert the comments to several vectors/representations which represent an opinion towards the topic. When large enough numbers of comments are collected, the system will start to analyze the comments. The opinion collected will be stored in vector space when training the model. Since similar opinions will be close in the vector space, the system should know the popular opinions and the system can get and show the popular opinions by directly getting the original sentence or generating sentences from a generative model. Scores will also be given to each popular opinion for representing if the public is more positive or negative toward the issue. Public sentiments can therefore be derived from the data.

2.2.1 Autoencoder and Variational AutoEncoder (VAE)

Since there is no available dataset for mapping sentences to vectors, the model will have to be trained in an unsupervised way. The autoencoder makes use of unsupervised learning, where the learning model is trained by unlabelled data and the features are obtained from the data (Ng, 2011). Figure 2.1 shows the architecture of an autoencoder.

Figure 2.1 Architecture of an autoencoder. L1 is the input layer, L2 is the hidden layer and L3 is the output layer. The $x_1$ to $x_6$ represents the input to the autoencoder. The $+1$ in L1 and L2 is a bias vector used during the training of the model (Ng, 2011)
An autoencoder is an architecture that contains an encoder and a decoder. In figure 1.1, the encoder is between L1 and L2 while the decoder is between L2 and L3. The encoder takes an input and outputs a latent vector representation. The decoder takes a latent vector and gives an output. There can be multiple hidden layers between L1 and L2, as well as between L2 and L3. The training objective of the autoencoder is to reconstruct the input given the encoded latent vector. In the case of this project, the autoencoder’s output is a sentence that has a similar meaning as the input. The difference between the variational autoencoder (VAE) and the ordinary autoencoder is that the deterministic encoding function of the ordinary autoencoder is replaced by a learned posterior recognition model (Bowman et al., 2015). It models the latent vector as a Gaussian distribution instead of only a single vector in the latent space. This can probably make the latent space to be more meaningful.

2.2.2 Simple Contrastive Learning of Sentence Embeddings (SimCSE)

The training objective of SimCSE is to make the representations of sentences with similar meaning to be more alike while making the representations of the other sentences to be more distinct (Gao et al., 2021). SimCSE directly uses large pretrained models such as BERT and RoBERTa, which improved BERT with some modifications and optimizations (Liu et al., 2019), to train under the contrastive learning objective.

During the training process, the unsupervised SimCSE is trained by only considering the sentence itself with dropouts as positive instances, while considering other sentences as negative instances, regardless of their meaning. There are also different dropout masks in the two forward processes. For supervised SimCSE, it is trained using the Stanford Natural Language Inference (SNLI) Corpus. The corpus contains paired text with human labels of entailment, neutral and contradiction. Paired entailment text pairs are considered as a positive instance, while the contradiction text pairs are considered as a hard negative instance and the other sentences as negative.

2.2.3 Discussion

Since the main task for our deep learning model is to represent an opinion in an n-dimensional vector space, where n would be the hyperparameter which can be adjusted, it is hoped that the model can get the high-level idea from a sentence and represent it in a vector. And for the reason that there should be no available dataset for mapping sentences to abstract ideas or vectors, this project will use an unsupervised way to train our model for getting the
representations of opinions in sentences. Autoencoder or VAE architecture would be a good choice since it is self-supervised and thus does not require labelled data.

Instead of using the basic settings in VAE, there can also be some adjustments in the model. Oshri & Khandwala (2015) proposed that besides the reconstruction loss which is calculated from the generated output with the input using cross-entropy, one can use additional encoder loss which compares the hidden representation of the input and generated output so that the meaning of the input and generated output is similar at least to the model itself. This can probably help a lot for the training process when using a higher ratio of reconstruction loss at the beginning of the training, and then increasing the ratio of the encoder loss. There are also many variants of the autoencoder in the area of NLP, such as using recurrent neural network (RNN), bidirectional RNN and convolutional neural network (CNN) as the encoder or decoder (Oshri & Khandwala, 2015). We will be examining different variants and trying to find the best combination. Some pre-trained language models like Bidirectional Encoder Representations from Transformers (BERT) may also be incorporated into our model since the large pretrained language model already can somehow get the meaning of a sentence. Besides using BERT, the pretrained models from SimCSE can also be used as the initialization of the parameters since the training of SimCSE aims to learn a better representation of sentences as compared to the pretrained models like BERT.

For getting the sentiment, the system will be using a method like transfer learning in the field of machine learning. Radford et al. (2017) also used an unsupervised way to train a Long short-term memory (LSTM) model for next character prediction using a corpus of 82 million reviews from Amazon. They found a sentiment neuron that is highly correlated to the sentiment of the review. Therefore, since sentiment is probably related to the high-level meaning of a sentence and would affect the output of the decoder, many public datasets for sentence sentiment can be used to train a simple additional multilayer perceptron which maps the opinion representation into sentiment score. After the sentiments of the popular opinions are obtained, the model can estimate the total sentiment score toward certain issues with the scores of the individual opinions and the number of likes or upvotes.

2.3 Evaluation

At the very beginning of the training, the performance of the model may need to be checked manually and the model will be adjusted accordingly so that it will be trained in the
correct direction. In the later stage of the training, the performance of the trained model will be benchmarked using various evaluation datasets and metrics. A bilingual evaluation understudy (BLEU) score is used for evaluating the performance of the reconstruction task (Oshri & Khandwala, 2015). IMDB review sentiment classification dataset and Microsoft Paraphrase Corpus are also used for evaluating the performance in sentiment analysis (Radford et al., 2017).

2.4 User Interface for Result Delivery

A webpage will be used for delivering the analysis result to the users for ease of development and accessibility. React.js and Node.js will be used for building the website as they are two of the most used programming languages nowadays for building websites. They bring great utility and convenience when building and editing the website. The webpage will show a list of popular keywords, alongside each keyword, there will be a list of the most popular and common opinions from the general public and a sentiment score for showing if the majority of the opinions are positive or negative towards the issue. The data can be updated on a daily or weekly basis after the machine learning models are trained and the pipelines are programmed. There will also be a history page for showing the history data.
3. Current progress and preliminary result

Section 3.1 shows the current progress of the data collection part of this project. Section 3.2 talks about the work done currently for the model architecture. Section 3.3 discusses the preliminary results of the model training. Section 3.4 talks about the limitations of the project.

3.1 Data Collection

Two million tweets and over a million Reddit posts were retrieved using the APIs. Since there are some bots in these social media platforms upload duplicates post, the duplicated Tweets, Reddit posts and comments were removed. For better training purposes, the unwanted parts such as hyperlinks, hashtags and user mentioning were also removed. Since the official API only allows to collect 500,000 Tweets per month, another library, sns scrape, is also used to collect more Tweets for future training and daily data collection purposes.

3.2 Work Done for Model Architecture

Currently, this project has made use of VAE and SimCSE as the model architecture. BertTokenizer is used to tokenize the words in the sentences into tokens, which are served as the input for the language model. Several modifications were made to the third-party library code which implements the sentence VAE. We changed the dataset to the data collected by the data collection pipelines from Twitter and Reddit. Also, the LSTM neural network was added and used as the encoder and the decoder, which allows us to compare the performance of different types of RNNs. Moreover, activation functions were added to the model to increase its expressiveness of the model. Furthermore, the code was edited to prevent information leakage at the RNN decoder due to the bidirectional RNN. In the original code, teacher forcing is used as the input of the decoder, which uses the ground truth as the decoder input, we changed the input of the RNN decoder to be the previous output of the decoder to have better training. Lastly, some visualization code is added to better visualize the loss and the performance of the model.
3.3 Preliminary Result for the Model Training

3.3.1 Vanishing KL Divergence Loss

During the early phase of the model training, the model suffered from the problem of vanishing Kullback–Leibler (KL) divergence loss. KL divergence loss in VAE penalizes the difference between the latent vector distribution and normal Gaussian distribution so that the space is more regularized and potentially more meaningful. The KL loss of the model drops to zero after a few update steps. Bowman et al. (2015) stated that when the KL loss is zero, the decoder ignores the latent code from the encoder and generates the output sentence directly. From the later investigation of the latent code distribution, the latent code generated by the encoder is in normal Gaussian distribution.

3.3.2 Visualization of the Latent Vectors

In order to get an idea of how the models are performing, we visualized the latent code in different ways.

First, from the SNLI corpus, text pairs with the three different labels can be obtained, some pairs do not have labels since the five human judgements cannot make a consensus. For each paired text in the test set, the corresponding latent codes are encoded and the cosine distance of the paired latent codes is computed. The box plots of different RNN models are shown in Figure 3.1.
Different RNN models are trained under the VAE settings and bidirectional RNN is used in the encoder to better capture the meanings in the input sentence. In Figure 3.1, there is no significant difference between the cosine distances distributions of different labels for all the models, and the cosine distances are normally distributed and centred at 1. For a good encoder that is trained from VAE, we expect the encoder to produce similar latent codes for sentences with similar meanings, which are the entailment pairs, and produce very different latent codes for the contradictive sentences. This shows that the current RNN models cannot get and encode the semantic meaning of the sentences. We then use the pretrained models from the SimCSE to plot the same graph, which can be seen in figure 3.2.
As shown in Figure 3.2, the sentence embeddings produced by SimCSE are better than our RNN models since the cosine distance is closer for entailment pairs and farther for contraction pairs. But there is overlapping between the cosine distance distributions which means it would not be very accurate if only the cosine distance of the embeddings is considered when deciding if two sentences are semantically similar.

To evaluate the performance of different models in our data, we use Reddit comments for the key phrase “Top Gun: Maverick” with depth only 1, which are the comments that are replying to Reddit submissions but not replying to other comments. We filtered the comment depth so that the comments obtained would be more relevant to the topic instead of focusing on the local discussions. This set of comments contains 1552 sentences after removing duplicates. The distribution of the embeddings is visualized after dimension reduction using Uniform Manifold Approximation and Projection (UMAP), which is an efficient dimension reduction technique that can preserve both the local and global features (McInnes et al., 2018). Figure 3.3 showcases the distributions of the embeddings of the two models.

![Figure 3.3 Latent codes distributions after UMAP of the two models](image)

The distributions shown in Figure 3.3 is similar to those distributions in Figure 3.1 and 3.2, the RNN model trained from scratch gives normally distributed latent codes which contains no meaning and the unsupervised SimCSE without any finetuning gives a more reasonable distribution since some sentences are similar in meaning and some are different. Since the graphs in Figure 3.3 show that some sentence embeddings are close, we use a \( k \)-dimensional tree, which is a kind of binary space partitioning tree data structure for \( k \)-dimensional points, to organize all the latent codes from each model and efficiently search for similar latent codes.
given a latent code of a sentence. We can therefore see if the sentences with similar latent codes are really similar in meaning, which can be seen in Figure 3.4.

![Figure 3.4](image)

(a) VAE with RNN

(b) Unsupervised SimCSE using pretrained BERT

Figure 3.4 The search results of similar embeddings with the input sentence from different models. For each of the results, the first value is the cosine distance, the second value is the index in the testing set and followed by the original sentence of the embedding.

Figure 3.4 shows some sentences with similar embeddings given a sampled sentence with a threshold of cosine distance 0.025, some results in Figure 3.4(b) are not shown. We can see that the searched sentences usually contain “Top Gun” and “Maverick”, but most of them are not having similar meanings to the input. The pretrained model should be finetuned to suit our use.

3.4 Limitations

One of the limitations of the system is that the language model will be trained by unsupervised learning. For unsupervised learning, the model learns from unlabelled data, so the model may not perform as well as supervised learning.

Another limitation is the Tweets and Reddit posts are collected according to different keywords presented in the trending news from Google Trends. We observed that some of the data collected may be irrelevant to the actual trending news. Moreover, since only one keyphrase is currently used per issue, some relevant posts are missing from the keyword search. More phrases related to the issue provided by Google Trends can be used for searching.
4. Future Works

4.1 Project Schedule

Table 4.1 showcases the renewed schedule for this project.

<table>
<thead>
<tr>
<th>Time Periods</th>
<th>Tasks</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sep 2022</td>
<td>Setting up the pipeline for collecting data from Twitter, Reddit and Google Trends</td>
<td>Completed</td>
</tr>
<tr>
<td>Late Sep 2022 – Feb 2023</td>
<td>Collecting and cleaning the data for training different language models</td>
<td>In Progress</td>
</tr>
<tr>
<td>Mid-Oct 2022 – Mid-Feb 2023</td>
<td>Experimenting and reviewing different deep learning models</td>
<td>In Progress</td>
</tr>
<tr>
<td>Late Dec 2022 – Jan 2023</td>
<td>Training and testing the initial model</td>
<td>Completed</td>
</tr>
<tr>
<td>Feb 2023 – Mid-March 2023</td>
<td>Building and designing the website for displaying the results</td>
<td>Pending</td>
</tr>
<tr>
<td>Late Feb 2023 – Mid-April 2023</td>
<td>Finalizing the language model</td>
<td>Pending</td>
</tr>
<tr>
<td>Mid-April 2023</td>
<td>Deploying the webpage</td>
<td>Pending</td>
</tr>
</tbody>
</table>

Table 4.1 Renewed Project Schedule

Currently, we are behind the original schedule since we have faced some unexpected difficulties in the task of experimenting and reviewing different deep learning models. Hence, we extended the task until mid-February because it is the most important part of the project.

In addition, it will take around 1.5 months to design and build the webpage for displaying the results of our project. By late February, the language model will be finalized after comparing the performance of different models, and we will train our model using the data collected from late September. In mid-April, the webpage will be deployed by either hosting it locally or renting a virtual machine to host the website so that the website can be accessed publicly.

4.2 Possible Improvements

We have thought of a few ways to improve our model. Firstly, a learning rate scheduler may be used to adjust the learning rate to potentially make the training more effective. Secondly, different parameters will be tried for the optimizer to try to lower the loss. Thirdly, different anneal functions such as the cyclical annealing schedule for the KL weight to achieve better convergences. Lastly, pre-trained transformers such as BERT and SimCSE may also be incorporated into the model.
5. Conclusion

Social media plays an important role in the current society, from sharing daily life to giving opinions about popular topics and issues. While it is important to perform sentiment analysis to understand the thoughts of the public, it takes a huge amount of effort and time to gather and analyze the opinions of the public. The purpose of this project is to build an automated sentiment analysis system on the more general popular topics. Users and researchers can thus understand the public’s opinions and attitudes on popular issues without having the need to look through a large number of comments and posts on social media platforms. The system can also be further extended for the use of other groups of interest.

The system collects data from APIs, such as Twitter and Reddit because the news is spread at a different speed and has a different lifespan on the two platforms (Priya et al., 2019). The system also uses deep learning and machine learning models to summarize and analyze the data. Autoencoder is currently used as the architecture for the language model because the autoencoder can be trained in an unsupervised way. Different variants for the encoder and decoder in the autoencoder will be examined using the BLEU score and the best combination will be chosen. A website will also be built and deployed to showcase the result of the project. The content on the webpage will be updated on a daily or weekly basis.

Currently, two million Tweets and over a million Reddit posts have been collected and cleaned. The result of data collection is acceptable even though there are still some flaws such as it contains some unrelated data. Moreover, the performance of the model is currently not as good as expected as it makes use of unsupervised learning instead of supervised learning and it is difficult to train from scratch.

For future planning, the renewed project schedule will be followed to ensure the progress of the project is on track. Multiple potential improvements may also be adopted to increase the performance of the model. Some pre-trained models such as BERT may also be incorporated into the model. We will try our best to improve and finish the model as soon as possible. Research and testing will also be conducted so as to solve and overcome the difficulties and limitations encountered. For the next phase of the project, a webpage will also be designed and built to showcase our results.
References


Appendices