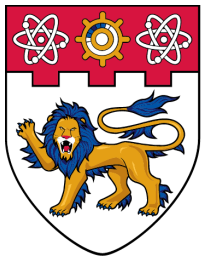


NANYANG TECHNOLOGICAL UNIVERSITY
SINGAPORE

CZ4034 Information Retrieval

Name	Matriculation Number	Contributions
Do Xuan Long	U1940040E	Q1 + Q5 + Data labelling
Do Duc Anh	U2120470F	Q4 + Data labelling
Goh Peng Aik	U2022363E	Q2 + HTML Code for front end + Data labelling
Lumlertluksanachai Pongpakin	U2023344C	Q2 + Python script for back end + Indexing + Data labelling
Jameerul Kader Faizan	U2023863D	Q3 + Data labelling + Slides + Video Presentation

Question 1: Explain and provide the following

1. How you crawled the corpus (e.g., source, keywords, API, library) and stored it

Subreddit	politics	TIHI	MadeMeSmile	therewasanattempt	Total
# comments	7001	2895	2767	4457	17120
# posts	426	699	827	751	2703

Table 1: The number of posts and comments crawled for each subreddit.

Our dataset, named **Reddit4034**, is created by crawling the posts from Reddit through Reddit API (https://www.reddit.com/dev/api/) and the PRAW (Python Reddit API Wrapper) library. Specifically, we choose 4 subreddits to crawl our the posts and comments: `politics`, `TIHI`, `MadeMeSmile`, `therewasanattempt`. Our motivation for choosing them is that through manual investigations, we expect that most of the comments from `politics` are emotionally

negative, from `MadeMeSmile` are emotionally positive, from `TIHI` are emotionally negative, and from `therewasanattempt` are emotionally positive and negative. Since we have to make the dataset as balanced as we can among classes, we opt for these subreddits.

For each post, we record its `Title`, `Post Text`, `ID`, `Score` and `Post URL`. We further crawl the comments from each post. Since we need to ensure that our dataset has "at least 10,000 records and at least 100,000 words", we filter out all the comments which has less than 10 words. Additionally, since the popular posts may have up to thousands of comments and we want our dataset to be a diverse dataset of comments, we limit the number of comments crawled up to 30 for each post. We present the number of posts and number of comments from each subreddit that we crawled in Table 1.

We store our dataset in google drive, and save another version of it on Firebase for hosting the interfaces.

2. What kind of information users might like to retrieve from your crawled corpus (i.e., applications), with sample queries

Our dataset contains the comments from 4 subreddits: `politics`, `TIHI`, `MadeMeSmile`, `therewasanattempt`, and brings rich information for a diverse sets of topics:

- `politics`: Users might want to retrieve information about current events, political opinions and discussions, and news related to political issues. This could include data such as the number of upvotes and comments on political posts, the sentiment of comments, the frequency of mentions of certain political figures or topics, and the overall engagement with political content on the subreddit.

- `TIHI` (Thanks, I hate it): Users might want to retrieve data on the types of content that are posted on the subreddit, such as images or videos that people find disturbing or unsettling. This could include data on the frequency and popularity of different types of content, the sentiment of comments, and the demographic information of users who engage with the content.

- `MadeMeSmile`: Users might want to retrieve data on positive and heartwarming content, such as stories, images, or videos that make people feel happy or inspired. This could include data on the most popular types of content, the sentiment of comments, and the demographic information of users who engage with the content.

- `therewasanattempt`: Users might want to retrieve data on humorous content related to failed attempts, such as images or videos of people trying to do something and failing. This could include data on the most popular types of content, the sentiment of comments, and the demographic information of users who engage with the content.

3. The numbers of records, words, and types (i.e., unique words) in the corpus

After crawling the raw data, we would like to filter our dataset such that it contains only high-voted (trending) posts, and make sure that all the comments are generated by humans and none of the two comments are the same. As such, we conduct the 2nd filtering step, in which we remove all the posts which have the scores less than 50. We then filter out all the repetitive comments, and all the comments that has hyper-links to remove the machine-generated comments. One such machine-generated comment is:

```
Welcome to /r/MadeMeSmile. Please make sure you read our [rules here.](https://www.reddit.com/r/MadeMeSmile/about/rules/) We'd like to take this time to remind users that:

We do not allow any type of jerk-like behavior, including but not limited to: personal attacks, hate speech, harassment, racism, sexism, or other jerk-like behavior (includes gatekeeping posts).

Any sort of post showing a mug, a shirt, or a print is a scam. You will not receive anything except a headache and a stolen credit card.

[More information regarding rule 1 as well as how mug/shirt/poster scammers operate can be found here.](https://www.reddit.com/r/mademesmile/wiki/rule1)
```

After the 2nd filtering step, we obtain our dataset, **Reddit4034,** which contains in total of 11388 comments. The number of final posts and comments are shown in Table 2. We observe that **Reddit4034** contains a large number of posts (2152) and each post contains a small number of comments kept, 5.29 on average.

Subreddit	politics	TIHI	MadeMeSmile	therewasanattempt	Total
# comments	5343	2669	1662	1714	11388
# posts	356	550	622	624	2152

Table 2: Statistics of the **Reddit4034** dataset.

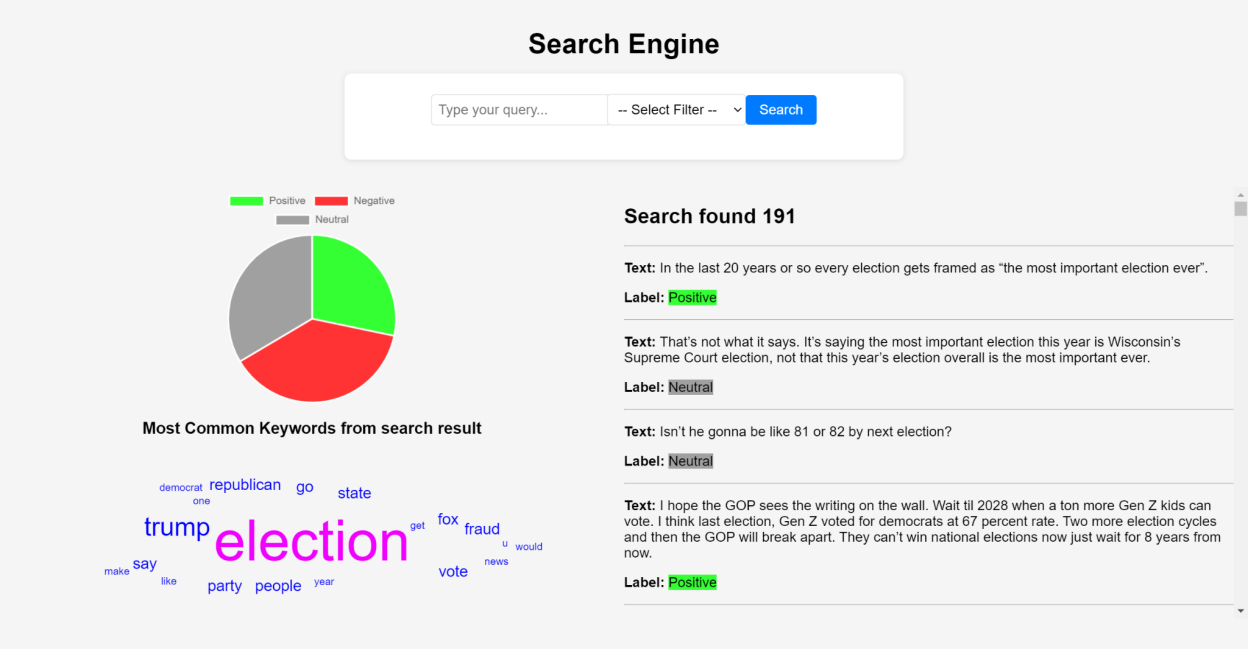
Finally, we report the linguistic statistics of our dataset in Table 3. We observe that each comment in **Reddit4034** contains on average of 33.70 words, and **Reddit4034** consists of a large number of unique words (46733) and unique tokens (27264), which indicates the linguistic diversity of our dataset.

# of comments	# of posts	Avg. # of words	Avg. # of tokens	# of unique tokens	# of unique words
11388	2152	33.70	39.53	27264	46733

Table 3: Linguistic statistics of the **Reddit4034** dataset.

Question 2: Indexing

For our web UI for the search engine, we used HTML, CSS and Javascript for the front-end, and Python for the back-end. The search engine consists of a simple search bar, a filter tab to filter our results by their sentiments (Positive, Neutral or Negative). The results are presented in row form in the format of the text followed by sentiment.



Our search engine is primarily based on TF-IDF (Term Frequency-Inverse Document Frequency) methodology. After completing the data crawling process as discussed in the previous section, the next step is to clean the data. Data cleaning involves several steps. Firstly, all the text in our dataset is converted to lower case to ensure consistency. Next, we perform word tokenization, which involves splitting the sentences into individual words. Subsequently, we apply word lemmatization to reduce words to their root synonyms. Lastly, we remove stop words, which are commonly occurring words, to obtain the root keywords for each text. The examples of the original text and the keywords obtained after data cleaning are depicted in the accompanying images.

```
"You're telling me that the wild claims of voter fraud that Trump was making up on the spot weren't credible? Well now I don't know what to believe!"
```

```
'tell wild claim voter fraud trump make spot credible well know believe'
```

In the TF-IDF process, the first step involves obtaining input queries from users. Subsequently, we calculate the term frequency-inverse document frequency (TF-IDF) of the user input with respect to our keyword corpus. Thereafter, we employ cosine similarity to rank the documents based on their relevance to the user queries. Finally, we present the documents to the user, ranked according to their cosine similarity score with the input queries. As an example, the appended image displays the search results obtained using the query 'Trump and Biden'.

```
Found 690 matching documents:
It's really sad Biden and Trump are all we have on offer from either party
-1.0

It's a fine strategy. Trump will run an incredible dirty campaign against anyone who runs against him (see: DeSantis groomer accusations).

If Trump emerges, Biden's focus on Trump will work out great and he can really pump up the whole "I beat Trump already" side of things.

If Trump loses the primary, it will be after a very mud slinging campaign and whoever wins will come out bruised and a lot of Republicans will be upset about Trump losing and not vote at all. Biden can focus on his opponents lack of experience (the go to move for the old guy) and highlight saving social security and Medicare (doesn't even matter if this is true, he will campaign on it) and get a decent chunk of the elderly vote.

Biden is a good politician
0.0
```

The five queries chosen for our example include 'get', 'work', 'mind', 'portal', and 'Trump and Biden'. These queries were selected based on their frequency in our corpus. 'Get' is the most frequent keyword, appearing over 2000 times. 'Work' appeared approximately 500 times, while 'mind' is found around 100 times. 'Portal' only appears once in our corpus. 'Trump and Biden' is a phrase search with a combined frequency of approximately 1000 times, which we aim to compare with simple search results. The table below presents the search times for each of the queries.

Query	Search Time (in sec)
‘get’	0.026996
‘work’	0.011984
‘mind’	0.006001
‘portal’	0.005001
‘Trump and biden’	0.015003

Question 3:

By including data visualisation tools such as a pie chart, it easily showcases the percentage of the labels that the search results have captured. Pie charts are particularly useful when you want to show how different parts contribute to a whole, as the chart's circular shape allows for a clear comparison between the sizes of each data point. This can be especially helpful when dealing with large sets of data that may be difficult to understand when presented in a table or text format.

We initially took a pie chart design from the internet and worked to implement that into our html code. So the initial design of figure 1 was worked to fit our code and resulted in Figure 2.

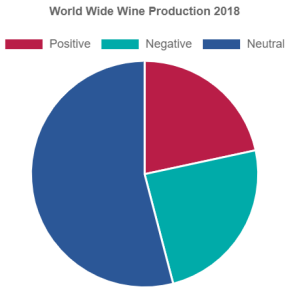


Figure 1

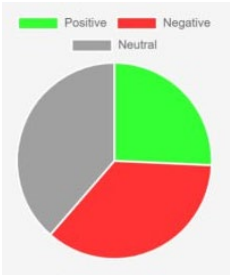
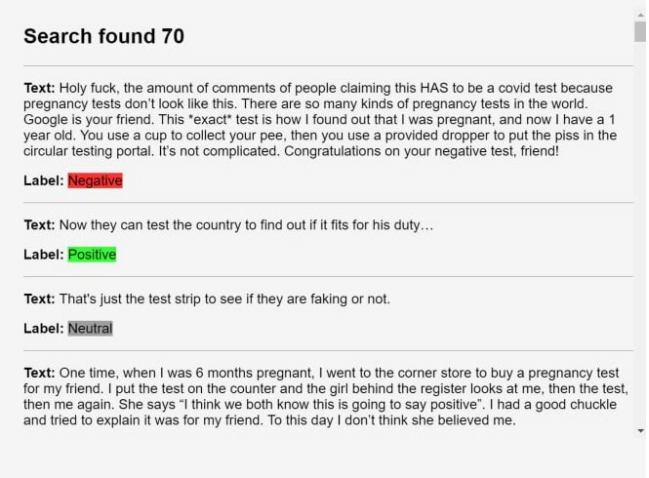


Figure 2

Also to better visualise and associate the labels of the pie chart (positive,negative or neutral) with the texts from the search results, we coloured each individual result to the corresponding colours we felt suited the labels and the emotions better (positive - green, negative - red, neutral - grey).

Figure 3



Next, we wanted to visually represent the most frequently used words in the given text of the search results. We decided to go with word clouds because they offer a quick and easy way to communicate the key themes or ideas within a large body of text. By using different font sizes, colours, and styles, word clouds can emphasise certain words or phrases that are particularly relevant or important.

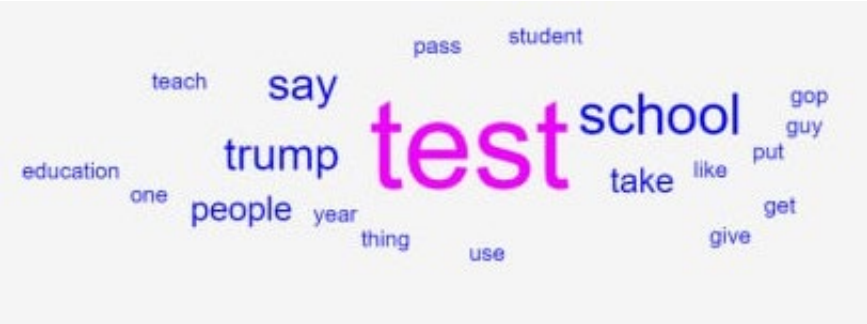


Figure 4

With that, we've come to the end of data visualisation tools that we used in our search engine to communicate the complex information and patterns in an intuitive and compelling way. By representing data visually, patterns, relationships, and trends can be more easily identified and understood, even by those without specialized training in the field. This can be particularly useful for decision-making, where large amounts of data can be difficult to comprehend and draw meaningful insights from.

Question 4 (Classification) : Perform the following tasks

- Motivate the choice of your classification approach in relation with the state of the art
    - In our work, we leverage the latest advances in machine learning based on the Transformer architecture and employ the Bidirectional Encoder Representations from Transformers (BERT) model as the basis for our classification method. To tackle the Binary Classification challenge of subjectivity classification (neutral and emotional) and toxicity classification (positive and negative), we fine-tune the BERT model using our own dataset. This approach allows us to take advantage of the powerful pre-trained representations provided by BERT and adapt them to our specific classification problem.
- Discuss whether you had to preprocess data (e.g., microtext normalization) and why
    - To prepare the crawled text data **Reddit4034** from Reddit for classification, it is necessary to preprocess each post to create a predictable and analyzable form for our classification task. Given that **Reddit4034** is often noisy with hashtags, emojis, and teencode and shortened words, preprocessing and cleaning are necessary to standardize the data for our Classifier, enabling it to learn the training data effectively without being impacted by noise.
    - We manually filter out the sentiment of the text, i.e., whether it is positive, negative, or neutral.
    - We remove any unnecessary characters, such as URLs, special characters. We keep emojis since they might have some useful information in determining the sentiment of a sentence.
    - We filter out all the comments which have less than 10 words. Additionally, since the popular posts may have up to thousands of comments and we want our dataset

to be a diverse dataset of comments, we limit the number of comments crawled up to 30 for each post.

- Build an evaluation dataset by manually labeling 10% of the collected data (at least 1,000 records) with an inter-annotator agreement of at least 80% (it is recommended to have 3 annotators, but 2 is also OK).
    - Our dataset **Reddit4034** consists of 11388 samples, which are comments. We first divide among us to label the 11388 samples with one of three labels: -1 (negative), 0 (neutral) and 1 (positive). In total, we obtain 3940 negative samples, 4104 neutral samples, 3344 positive samples.
    - We further divide our **Reddit4034** dataset into 2 smaller datasets, which either contain labels (0 vs 1, -1) for subjectivity classification, called subjectivity dataset and (-1 vs 1) for toxicity classification, named toxicity dataset.
    - We then divide the subjectivity classification dataset into train, valid, test splits by the ratio 80-10-10, resulting in 9106 training samples, 1137 validation samples, and 1140 testing samples.
    - We further divide the toxicity classification dataset into train, valid, test splits by the ratio 80-10-10, resulting in 5823 training samples, 727 validation samples, and 729 testing samples.
    - To make sure that our testing samples of subjectivity classification dataset are of high quality, we ask 3 annotators who are English native speaker to re-label 1137 testing sample. We then ask another expert annotator to accumulate the labels from three prior annotators of 1137 testing samples to create the final 1137 testing samples. We measure the annotators' agreements by Krippendorff's Alpha. Our annotators for subjectivity classification dataset achieves a high annotator agreement of 0.78. We apply the same method to verify our 1137 testing samples of the toxicity classification dataset, and our annotators achieve a Krippendorff's Alpha of 0.72.
- Provide evaluation metrics such as precision, recall, and F-measure on such dataset
    - In order to evaluate how well the two classifiers are able to classify the text data for subjectivity classification and toxicity classification, we will be using F1 score, precision score and recall score to evaluate the ability of the classifiers to make predictions for each of the class labels.
    - Precision is the proportion of true positive classifications out of all positive classifications made by the model. In other words, it measures how accurate the model is when it predicts a positive class. Mathematically, precision can be defined as: precision = true positives / (true positives + false positives)
    - Recall, on the other hand, is the proportion of true positive classifications out of all actual positive instances in the dataset. In other words, it measures how well the model is able to identify positive instances. Mathematically, recall can be defined as: recall = true positives / (true positives + false negatives)
    - F1 is a metric used to evaluate the performance of a classification model. It is a weighted average of the precision and recall of the model. The F1 score is calculated as: F1 = 2 * (precision * recall) / (precision + recall). The F1 score ranges from 0 to 1, with higher values indicating better performance.

    - Accuracy is defined as the proportion of correct predictions made by the model out of all predictions made. It can be defined as: accuracy = (true positives + true negatives) / (true positives + true negatives + false positives + false negatives)
- Perform a random accuracy test on the rest of the data and discuss results
    - We train our BERT-base-uncased model for binary classification on subjectivity classification dataset and subjectivity classification dataset with a batch_size of 8, max window length of 512, on 15 epochs with the learning rate of 5e-6. We use Adam as our optimizer with 500 warm-up steps.
    - Our model achieves an accuracy for the subjectivity classification test of 68.87% and an accuracy for the toxicity classification test of 74.83%.

```
{'eval_loss': 0.5992587208747864,
 'eval_accuracy': 0.6886543535620053,
 'eval_f1': 0.6537496731036572,
 'eval_precision': 0.6591981884289086,
 'eval_recall': 0.6507145972422586,
 'eval_runtime': 14.1166,
 'eval_samples_per_second': 80.544,
 'eval_steps_per_second': 10.13,
 'epoch': 15.0}
```

```
{'eval_loss': 0.5506489872932434,
 'eval_accuracy': 0.7482806052269602,
 'eval_f1': 0.7388716056409904,
 'eval_precision': 0.758480128715499,
 'eval_recall': 0.7375459215560738,
 'eval_runtime': 8.3087,
 'eval_samples_per_second': 87.499,
 'eval_steps_per_second': 10.952,
 'epoch': 15.0}
```

- Discuss performance metrics, e.g., records classified per second, and scalability of the system.
    - Subjectivity classification:
        - Train runtime: 5990.6402s
        - Records classified per second: 23

```
metrics={'train_runtime': 5990.6402, 'train_samples_per_second': 22.801,
```

    - Toxicity classification:
        - Train runtime: 3703.8384s

- Records classified per second: 24

```
metrics={'train_runtime': 3703.8384, 'train_samples_per_second': 23.582,
```

- Design a simple UI for visualising classified data.
    - We present our visualisation about the relation between F1 and training loss and number of training steps in *Figure 1* and *Figure 2*. From the plots, we can observe that the training loss decreases and F1 score increases with the increasing number of steps, which means that our model is achieving a good performance, indicating that our model is "learning" right.

Figure 1: Model performance on subjectivity classification dataset. The graph represents the plot between the value of F1 and training loss according to the number of training steps.

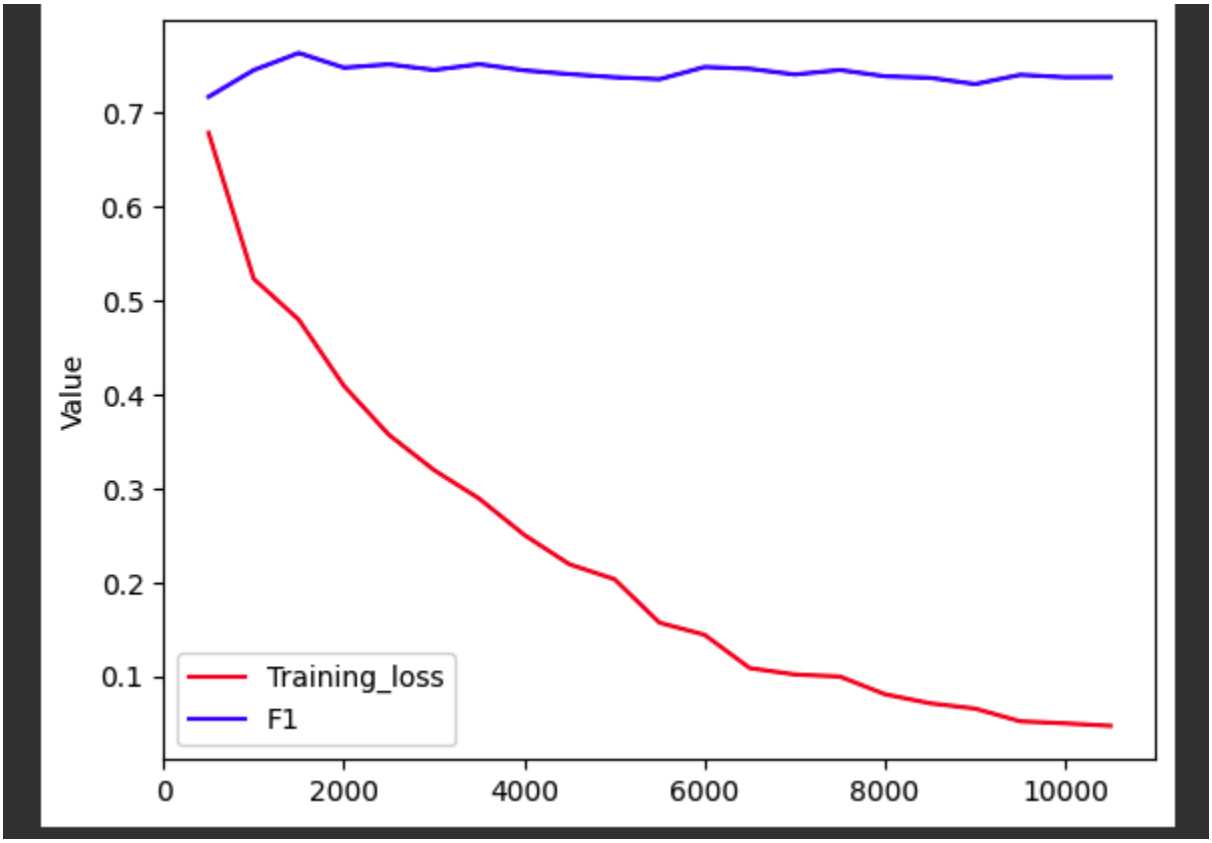


Figure 2: Model performance on toxicity classification dataset. The graph represents the plot between the value of F1 and training loss according to the number of training steps.

Question 5: Explore some innovations for enhancing classification. Explain why they are important to solve specific problems, illustrated with examples

In this section, we present two innovations that we have tried to enhance the classification capability of BERT.

Dataset	Neutral-Emotional				Positive-Negative			
Method	F1	Pre.	Rec.	Acc	F1	Pre.	Rec.	Acc
Enhancing	**66.15**	66.89	**65.78**	69.73	75.79	76.14	75.66	76.13
More data	66.01	**67.87**	65.44	**70.52**	**77.67**	**78.24**	**77.50**	**78.05**
Original	65.37	65.91	65.07	68.87	73.89	75.85	73.75	74.83

Table 5: Evaluation results of innovations. All metrics are measured in percentage.

**** Solution 1:** Enhancing the input comment by adding emotional words.*

Through manual examinations, we notice that the class of a comment strongly depends on the emotional words presented in that comment. For instance, given a positive comment "`Thank Florida for Rick Scott and Ron DeSantis and Matt Gaetz and a whole slew of the crazy train. Old people who fantasize about Ronny Reagan vote GOP despite how far down the luny bin the party has gone.`", *we observe that this comment has a number of positive words such as "`Thank`", "`crazy`", "`fantasize`", which determine the positive label of the comment.* Motivated by this, given an input comment, we attempt to pre-determine positive and negative words before feeding into the model. To do so, we use the predefined categories in the `movie_reviews` corpus in `nltk` library to create sets of positive and negative words. Given a comment, we tokenize the comment by `nltk.word_tokenize` and examine all the tokens if any of them in the set of `movie_reviews` positive or negative words. We then append all the positive and negative tokens to the end of the comment in the following format: "`Comment: comment. Positive words: positive_words, Negative words: negative_words`". Finally, we train the BERT model with the same setup as the original dataset, and we obtain the results shown in method **Enhancing,** Table 5. We observe that by adding positive and negative words to the end of the input comment, the model's performance in both tasks of subjectivity classification and toxicity classification is improved compared to the original method, which verifies the effectiveness of positive and negative words in helping the model classify the comments. This improvement also shows that our innovation in pre-determining positive and negative words in the comments to train the model is effective.

Some examples of the comments, which are wrongly classified by the original BERT, are correctly classified by method of **Enhancing** BERT are presented in Table 6 (the first two comments).

Input comment	Correct label	Original BERT	Enhanced BERT	More-data BERT
Wash your hands, boys and girls. ALWAYS WASH your hands, no matter what!	0	1	0	-
They already want 10 year olds to have babies, what's next? 9 year olds?	1	0	1	-
This is a good one for the guys who still think the DNC is compatible with Christianity.	1	0	-	1
And the house fire left no survivors. Welp, didn't need the house anyways.	0	-1	-	0

Table 6: Case studies in the task of subjectivity classification.

**** Solution 2:** Training with more data.*

We postulate that one possible method to boost the performance of BERT on our task is to train with more data. This is because our whole dataset contains only around 10K samples, which is relatively small for fine-tuning the BERT model compared with the number of parameters that the BERT model has (110M). To this end, we collect the Twitter and Reddit Sentimental analysis Dataset from Kaggle (cosmos98/twitter-and-reddit-sentimental-analysis-dataset), which has a similar data distribution with our dataset. Furthermore, this dataset is also about positive, negative and neutral comments, which is also aligned with us. This Kaggle dataset consists of 37249 samples with 13242 positive samples, 15830 neutral samples and 8277 negative samples.

Our strategy is to integrate this dataset into the training splits so that the model can be fine-tuned on more data for each task. To do so, we have to make sure that there are no samples in our test splits appearing in this dataset. As a result, we filter-out all the samples in this Kaggle dataset that are in our test splits for both tasks of subjectivity classification and toxicity classification, resulting in 0 samples removed. Finally, the statistics of our new datasets for the two above tasks are presented in Table 7.

Dataset	# Train	# Valid	# Test
subjectivity classification (neutral-emotional)	33213	1137	1140
toxicity classification (positive-negative)	43072	727	729

Table 7: Statistics of our new datasets for 2 classification tasks.

We train the BERT model with the same setup as the original dataset, and we obtain the results shown in method **More data,** Table 5. We derive two main observations. First, the **More data** version of BERT outperforms all other methods significantly on the task of toxicity classification, which consolidates our postulation and verifies the effectiveness of including more helpful data to train the model. Second, the **More data** BERT model achieves competitive results with the **Enhanced** method in the task of subjectivity classification, and even higher Accuracy score, which illustrates that adding more data is somehow beneficial to the model to be trained on this task.

Two last rows in Table 6 are the examples of the comments, which are wrongly classified by the original BERT, and are correctly classified by method of **More data** BERT.

Youtube Link : https://youtu.be/iFaWmY4xC4c

Submission Part 4:

https://drive.google.com/drive/folders/1_U2FgljTlv5w231ahcFdhCUDegaYmJyF?usp=sharing

Submission Part 5:
https://drive.google.com/drive/folders/1acYcb3hibUHgn_a_tCzdu9b278uVr2Yz?usp=sharing