BT4103

Business Analytics Capstone Project

Final Report

AY22/23 Semester 2

Team 3

Charmaine Chng	A0204142Y
Lek Ze Ning	A0203667A
See Cheng Jiang	A0218252L
Tay Zhi Sheng	A0201840W
Teo Jun Hao Bryan	A0204347L

Executive Summary

The Multi-lingual Text Classification project aims to develop a robust and efficient system that can automatically classify emotions and stances from texts in various languages, with respect to an entity of interest in the text. This system will help the host organization to streamline their research process, allowing them to gain better insights into social issues and trends.

Currently, the host organization faces significant challenges in the manual labeling of data, which is time-consuming, expensive, and unscalable. By leveraging deep learning models, our project aims to automate this process, significantly reducing the time and cost required for data labeling while improving the accuracy of the results. This will enable our host organization to maximize the potential of their data and enhance the quality of their research outcomes. In addition, a user-friendly interface will also be developed for users to upload and retrieve the labeled dataset.

Ultimately, we aim to provide our host organization with an efficient, cost-effective, and scalable solution that can enhance their research capabilities and provide valuable insights to stakeholders.

Acknowledgment

We are incredibly grateful for the unwavering support and guidance provided to us by our academic supervisor, Professor Wang Qiu Hong, our company supervisor, Mr Chen, and our Teaching Assistant, Joel Quek. We extend our heartfelt appreciation for their patience, invaluable advice, and extensive knowledge that enabled us to successfully complete this project. Without their tireless efforts and dedication, this accomplishment would not have been possible. Once again, we express our sincere gratitude to these individuals for their contributions to our project and our academic journey.

Table of Contents

1 Introduction

1.1 Background of Host Organization

Our host organization plays a pivotal role in enhancing the operational effectiveness and preparedness of the nation through the application of behavioral sciences.

1.2 Project Objective

The Multi-lingual Text Classification project aims to leverage deep learning to automate the identification of emotions and stances from texts in different languages, with respect to an entity of interest in the text. This streamlines our stakeholder's research process, enabling them to maximize the potential of their data and enhance the quality of their research outcomes. Thus, enhancing their research capabilities and gaining deeper insights into social issues and trends.

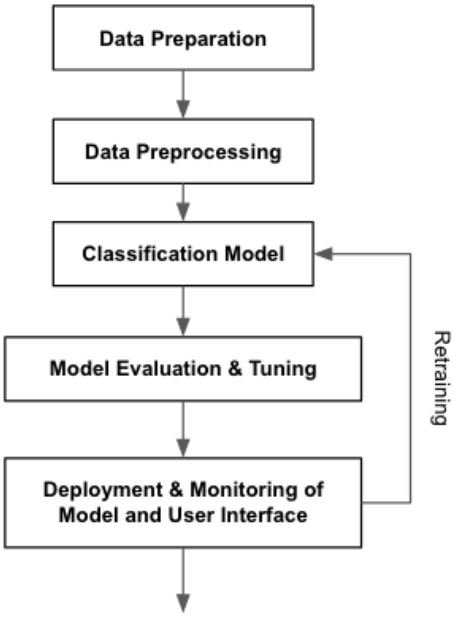


Figure 1: Project Flow

We adopted the machine learning lifecycle to approach the project (Fig 1) to ensure that the end solution would not only be delivered timely but also meet the project objectives. The following sections will be structured similarly to the machine learning lifecycle.

2 Analytical Requirements

The following analytical requirements act as a foundation for addressing the project objective.

2.1 Model Selection

In view of the tight project schedule, limited resources, and the limited amount of data we have, our group has opted to utilize pre-trained language models instead of training a model from scratch. Pre-trained language models can allow us to achieve the same or even better performance much quicker and with much less labeled data.

As the developed model will ultimately be used by our host organization and its stakeholders for a variety of project scopes, a pre-trained model is ideal for generalizability. To ensure this, we have researched and shortlisted state-of-the-art pre-trained models that have been trained on a large language corpus and have proven to be suitable for a variety of natural language processing (NLP) tasks. A deep dive into the model development and selection will be outlined in Section 5 of the report.

2.2 Model Evaluation

To evaluate the performance of the model, F1-score and accuracy will be measured and act as the baseline metrics.

The F1 score is a widely used metric in classification tasks that combines and balances the precision and recall of the model. Precision measures the proportion of correctly predicted positive instances among all predicted positive instances, while recall measures the proportion of correctly predicted positive instances among all actual positive instances.

We used macro F1-score and accuracy for model evaluation due to the imbalanced nature of the data, where some classes have significantly fewer instances than others. These metrics will be utilized in Section 5.2, in determining the best-performing model for the English and Chinese models respectively.

3 Functional Requirements

The following functional requirements specify the 3 deliverables of the project which are the model, the web application as well as the documentation, and ensures that the users' expectations are met.

3.1 Model

There are 2 separate models, each for the classification of English and Chinese texts for the 7 emotions and 3 stances. The 7 emotion classes are 'Joy', 'Surprise', 'Neutral', 'Sadness', 'Fear', 'Disgust' and 'Anger', while the 3 stance classes are 'Favor', 'None' and 'Against'.

3.2 Web Application

The web application should:

1. Have a file upload function for users to upload their unlabeled textual datasets. Data validation should be performed to ensure that only excel files of the specific format are permitted.
2. Generate descriptive statistics for the data that has been labeled by the model classifier, including, but not limited to, the following:
    a. Overall breakdown of labeled data by emotion and stance
    b. Breakdown of labeled data by emotion and stance for specific entities of interest selected through a filter function
3. Have a file download function to enable users to download the labeled datasets in excel format for their further analysis.
4. Ensure minimal code interaction for end users when performing their tasks.

3.3 Documentation

The documentation should:

1. List the steps taken to run the model (in case re-training is necessary)
2. Specify data requirements for the web application which are:
    a. The accepted file format is xlsx
    b. English/Chinese only data for English/Chinese only models
    c. Data files containing only the columns 'text' and 'entity'
3. List out steps to navigate the web application

4 Data

4.1 Exploratory Data Analysis

The dataset provided comprises 4 main columns which are the 'text', the 'entity' of interest, and the emotion and stance classes.

'text'

The 'text' column of the dataset was sourced from comments extracted by the host organization from various Singaporean social media pages, primarily frequented and commented on by locals, discussing current affairs and notable figures in the country. As a result, the text is often rife with Singlish expressions, colloquialisms, and emojis. It is important to note that words may be abbreviated or misspelled, adding to the complexity of the language used.

'entity'

The 'entity' column contains the specific entities of interest, which were extracted by the host organization's team using SpaCy's Named Entity Recognition. Since these entity terms are extracted directly from the text, any misspellings present in the text will be reflected in the extracted entities. It should be noted that the effectiveness of the entity extraction process is largely dependent on the accuracy and quality of the Named Entity Recognition model used.

'emotion' and 'stance'

The 'emotion' and 'stance' columns contain the target emotion and stance classes respectively, with respect to the entity of interest. The labels are based on the 7 psychological emotion classes (i.e., anger, sadness, disgust, neutral, fear, surprise, joy) and 3 stance classes (i.e., favor, none, against).

4.2 Data Preprocessing

Given that the data provided has been scraped from social media sources, it is inevitable that it contains significant noise. This will greatly impact the model's ability to learn effectively, leading to poor model performance. Since we will be leveraging a pre-trained language model which requires minimal data pre-processing (Devlin et al., 2018; E. Peters et al., 2018), we perform minimal but essential data cleaning for the purpose of reducing noise in the dataset.

For the English dataset, the steps taken to clean the data are as follows:

1. Removing rows with any column that is empty or contains non-English text
2. Removing rows where the 'text' or 'entity' column does not contain any alphabets, i.e. text contains only emojis, punctuations, and/or numbers

3. Removing rows where the ‘entity’ is a username, i.e. ‘entity’ appears at the beginning of the comment

For the Chinese dataset, the steps taken to clean the data are as follows:

1. Removing rows with any column that is empty or do not contain any Chinese text
2. Removing all non-Chinese text from the original text, except when the non-Chinese text is the entity itself
3. Removing rows where the ‘entity’ is a username, i.e. ‘entity’ appears at the beginning of the comment

4.3 Challenges

4.3.1 Small and Imbalanced Dataset

English Dataset

The initial English dataset had 915 data points and was highly imbalanced, with the ‘neutral’ emotion and ‘none’ stance classes representing more than 70% of the dataset. The data imbalance can result in a model that is biased towards the ‘neutral’ emotion class and ‘none’ stance class, leading to poor generalization and accuracy for the other underrepresented classes (Gupta & Gupta, 2019). Given that each data point is assigned to two labels, one for emotion and one for stance, addressing the imbalance distribution for both concurrently is a challenging task.

After conducting the manual labeling process, we observed a strong correlation between emotions and stances. Specifically, our observations indicate that positive emotions are more likely to be associated with a positive stance, while negative emotions are more likely to be associated with a negative stance. Given this insight, we have concluded that by targeting the imbalance emotion classes we will indirectly address the imbalance stance classes. By doing so, we believe that we can effectively improve the accuracy and generalizability of our model's predictions.

To address the data imbalance, we performed the following steps on our dataset:

1. To enhance the data points for the other emotion classes, we obtained additional unlabeled English data points from our host organization and manually labeled 1656 of them. While the ‘neutral’ class was still predominant in the newly labeled data, we were nevertheless able to boost the number of data points for the underrepresented classes. Hence, reducing the overall data imbalance of the emotion classes.
2. Undersampling of the ‘neutral’ emotion class by selecting only 200 ‘neutral’ data points to balance out all the emotion classes.

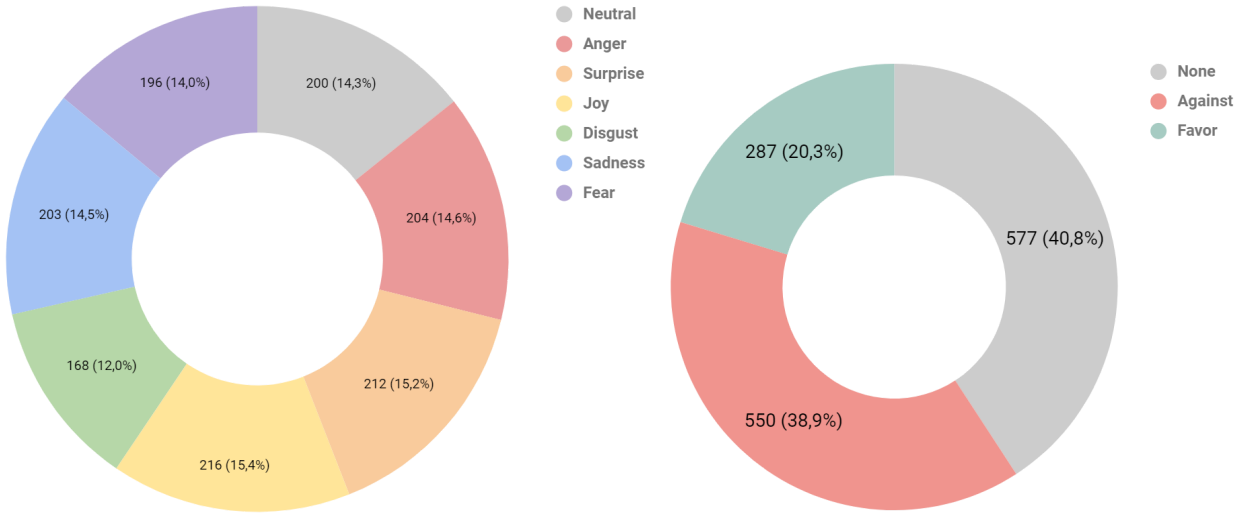


Figure 2: Distribution of final English dataset

As such, the final English dataset consists of a total of 1399 data points and is more balanced for the emotion and stance classes (Fig 2).

<u>Chinese Dataset</u>

The initial Chinese dataset was significantly smaller than the English dataset with only 395 data points and also faced the same data imbalance issue as the English dataset. Thus, following the same approach as the English dataset, we manually labeled and added an additional 794 data points to boost the number of data points and reduce the data imbalance. In addition, we also undersampled and selected only 100 'neutral' data points to further balance out all the emotion classes.

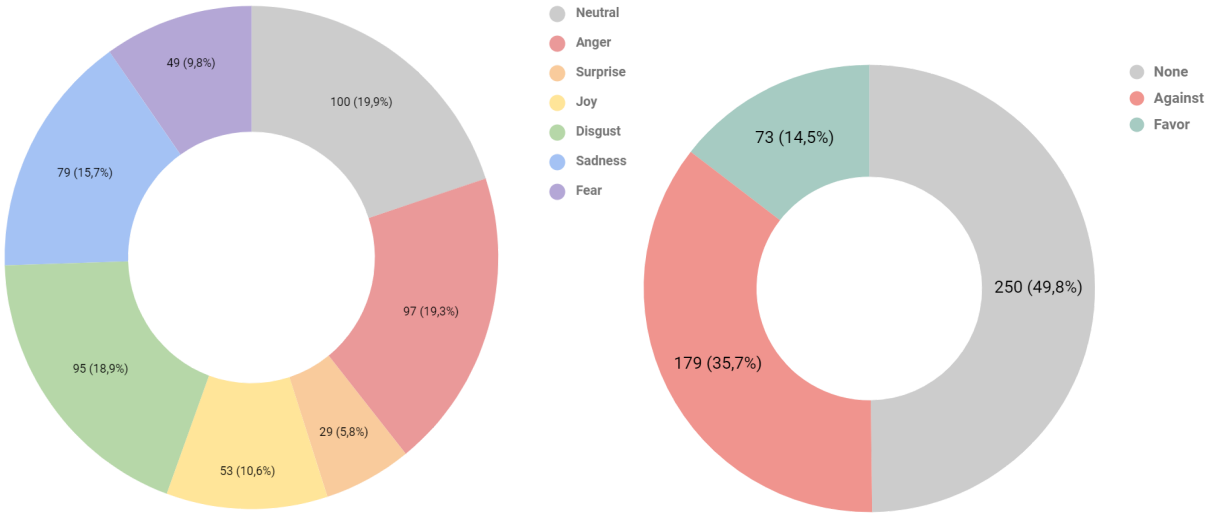


Figure 3: Distribution of final Chinese dataset

The final Chinese dataset consists of a total of 502 data points and is more balanced for the emotion and stance classes (Fig 3).

4.3.2 Entity-Based Sentiment Analysis

The main goal of the project is to identify emotions or stances towards an entity of interest mentioned in a text. This is a complex task as illustrated by the sentence:

"I love the food here, but the price is steep"

which contains two entities with opposite sentiment polarities. According to Jiang et al. (2011), about 40% of the below-average performance in sentiment classification is attributed to the lack of attention given to the target's context.

We transform the entity-based sentiment analysis task into a sentence-pair classification task through the construction of an auxiliary sentence similar to the next sentence prediction task used to pre-train BERT (Devlin et al., 2018). This enables us to leverage the pre-trained language model which has been demonstrated to be effective in minimizing the effect required for feature engineering while achieving excellent results (Devlin et al., 2018). The auxiliary sentence is constructed with only the entity of interest.

Baldini Soares et al. (2019) introduced a novel technique where they tagged entities with additional learned tokens before inputting them into BERT. Motivated by the paper's result, we added a custom-learned [ENTITY] token which replaces the entity of interest in the text and the auxiliary sentence. The choice of replacing the entity of interest with the [ENTITY] token not only helps to simplify the training process but also makes the model more generalizable. It also eliminates the risk of tokenizing entities not found within the pre-trained vocab with the [UNK] token, which is not meaningful for prediction.

4.3.3 Long Text Length

There is a substantial amount of data points in both the English and Chinese datasets with extremely lengthy texts that contain segments that are irrelevant in determining the emotion and stance of the entity of interest. As such, these texts, in their entirety, introduce significant noise into the data and feeding them directly into the model would complicate the training process and negatively affect the model performance.

In order to tackle this challenge, taking into consideration that opinion words are generally found nearer to the entity of interest, we employed the entity of interest as a reference point and extracted the words preceding and following it to form a maximum 50-word segment as the input text. This approach not only facilitates a more consistent training process, but also resolves the issue of exceeding the 512-token limit of the pre-trained language model. By selecting a fixed token length, we are able to avoid the difficulties associated with variable token lengths and ensure that our model is capable of handling a wide range of text length with greater accuracy.

5 Model

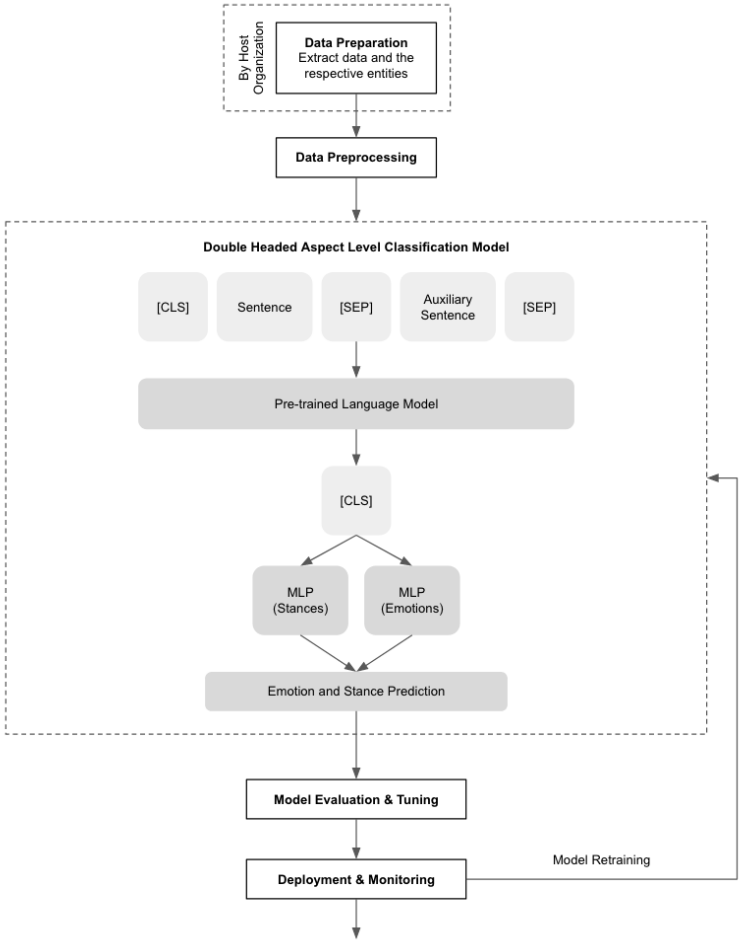


Figure 4: Overall Model Architecture

5.1 Pre-Trained Language Model

5.1.1 English

BERT

BERT (Devlin et al., 2018) is one of the most widely used and powerful encoder-based transformer models in the field of natural language processing (NLP). It is a pre-trained model that has been trained on a massive amount of text data, which has enabled it to capture the contextual relationships between words. One of BERT's key features is its bidirectional attention mechanism, which enables it to effectively analyze the input text from both directions. When fine-tuned for specific NLP tasks, such as sentiment analysis, BERT has consistently demonstrated state-of-the-art performance, making it a popular choice

for many NLP applications. Moreover, BERT's strong ability to handle domain-specific text, makes it an ideal language model of choice.

SingBert

SingBert[1] is an encoder-based transformer that has been initialized with BERT's weights and pre-trained on Singlish[2] and Manglish[3] text corpus collected from Reddit. The pre-training with a local text corpus makes SingBert highly relevant for processing English text data that contains a mixture of languages and dialects commonly used in Singapore. Nevertheless, it is important to note that the pre-training data is from a single source, which may introduce inherent biases due to differences in user base across social media platforms.

5.1.2 Chinese

ChineseBERT-WWM

ChineseBERT-WWM (Cui et al., 2021) is a highly effective variant of BERT that is specifically designed to tackle Chinese NLP tasks. It takes into account the unique characteristics of Chinese text, where characters are not constructed from alphabets, by masking entire Chinese words rather than individual characters. This approach enables the model to better understand and contextualize the Chinese language, capturing the intricate relationships between words in Chinese texts. Additionally, the language model is trained on a vast corpus of Chinese text that includes both simplified and traditional Chinese, making it a highly versatile and reliable tool for a wide range of Chinese language applications.

5.1.3 Loss Function

The choice of a loss function is vital in ensuring a performant model. Since the nature of the project is a classification task, we adopted and compared the model performance between cross-entropy loss and focal loss (Jiang et al., 2011) functions:

$$Cross\ Entropy\ Loss(\rho) = -log(p)$$

$$Focal\ Loss(\rho) = -(1-p)^{\gamma} log(p)$$

$p \in [0, 1]$ is the model's estimated probability for the target class and $\gamma \geq 0$ is a tunable hyperparameter.

Focal loss differs from cross-entropy loss by its modulating factor $(1 - p)^{\gamma}$. The modulating factor down-weight easy examples and up-weight difficult examples. When a text is misclassified and p is small, the modulating factor is close to 1 and loss is not affected. As p approaches 1, the modulating factor decreases to 0 and the loss for the well-classified labels is down-weighted. The hyperparameter γ

[1] zanelim/singbert · Hugging Face
[2] Singlish is an English-based creole language spoken in Singapore
[3] Manglish is an informal form of Malaysian English with features of an English-based creole used in Malaysia

smoothly adjusts the rate of down-weighting for easy labels. When the γ increases, the effect of the modulating factor increases, and when γ = 0, the focal loss is equivalent to cross-entropy loss.

5.1.4 Double-Headed Classifier

The classifier plays a critical role in predicting the sentiment associated with a specific entity. Specifically, it utilizes the [CLS] final hidden state from the pre-trained language model as input, which contains the summarized representation of the entire text sequence. This representation is then fed into a double-headed classifier that is responsible for predicting both the emotions and stances related to the given entity. Our team opted for this approach due to the high correlation between emotions and stances, as previously mentioned in Section 4.3.1. This double-headed classifier approach consists of two separate multi-layer perceptrons, each responsible for predicting a specific sentiment. The combination of these two models through a joint loss function reinforces the training process, leading to improved model performance. By using this approach, our model can provide a more nuanced understanding of the sentiment associated with each entity.

5.2 Evaluation

	English	Chinese
Training and Test Sets	Train Test Split Train Set: 1272 Test Set: 142	K-fold Cross Validation (10 folds) Train Set: 452 Test Set: 50
Dropout	0.35	0.35
Batch Size	64	32
Epochs	10	10
Optimizer	AdamW (Loshchilov & Hutter, 2017) Learning Rate: 2e-5 Weight Decay: 0.01	AdamW (Loshchilov & Hutter, 2017) Learning Rate: 2e-5 Weight Decay: 0.01
Learning Rate Scheduler	Exponential Gamma: 0.9	Exponential Gamma: 0.9

Table 1: Hyperparameters Used

5.2.1 Loss Function

The results outlined in the following table support our decision of utilizing focal loss. The comparison was made on the English dataset. It is observed that models trained with focal loss achieved better performance in the minority and difficult classes. Thus, highlighting the effectiveness of the modulating factor in the focal loss. We adopt focal loss across all the model training.

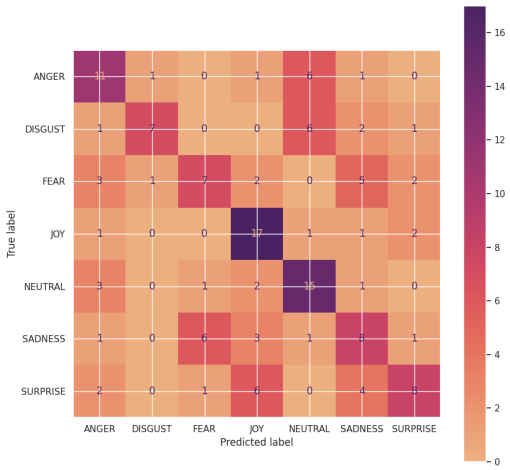
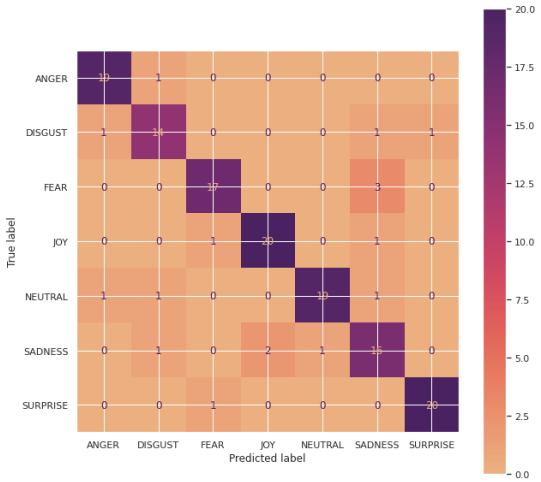
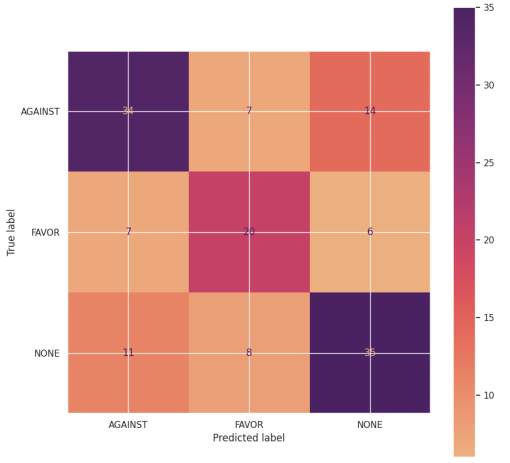
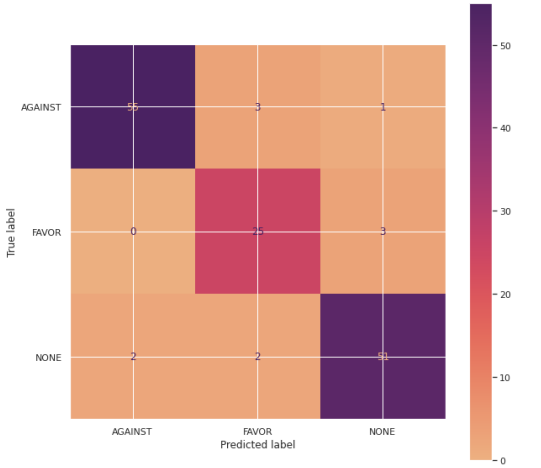
	Cross-Entropy Loss	Focal Loss
Emotions		
Stances		

Table 2: Labels Sentiment Heatmap

5.2.2 Language Model

Precision, recall, and F1-score are the metrics used for evaluating the models. In our use case, both precision and recall are important as false positives and false negatives can dramatically impact the way users interpret the predictions and ultimately the outcome of their research. Hence, we ultimately look

at both the individual and macro F1-score, which takes into account both precision and recall, to evaluate the model's performance. The individual prediction scores can be found in Section 13.2.

English Model	Emotion		Stance	
	Macro Accuracy	Macro F1-score	Macro Accuracy	Macro F1-score
BERT	0.88	0.88	0.92	0.91
SingBERT	0.63	0.62	0.70	0.70

Table 3: English Model Performance

Chinese Model	Emotion		Stance	
	Macro Accuracy	Macro F1-score	Macro Accuracy	Macro F1-score
ChineseBERT-WWM	0.74	0.75	0.89	0.88

Table 4: Chinese Model Performance

English

After comparing the results of the model with pre-trained BERT and SingBERT, it was found that the former outperformed the latter in terms of macro accuracy and F1-score by a significant margin (Table 3). This outcome was unexpected as SingBERT was believed to have an edge over BERT due to its Singlish-specific training and initialization with BERT weights.

We hypothesized that SingBERT's poor performance could be attributed to inherent bias resulting from its limited training data where it is only sourced from Reddit. As a result, it may not have captured the wide variety of ways Singlish is communicated, which could have affected its accuracy. Another potential factor could be the small model scale of SingBERT and its limited variety and size of training data, which could have led to the forgetting of BERT's language capabilities. In fact, Driess et al. (2023) found that the PALM-E model with 12 billion parameters experienced catastrophic forgetting of language capabilities. Nevertheless, when the PALM-E model scaled up to 562 billion parameters, it was found to result in less catastrophic forgetting. In the case of SingBert, even for the large model, it only has 340 million parameters.

Chinese

Stratified K-fold cross-validation was utilized to evaluate our Chinese model due to the limited Chinese dataset. This ensures that the results obtained are more robust and less likely by chance. Stratified K-fold cross-validation is a technique used to evaluate model performance by partitioning the dataset into K-folds while ensuring that each fold contains a similar distribution to the original dataset. The dataset was stratified on the emotion instead of the stance due to the more severe data imbalance faced by the

emotion classes. The results show that ChineseBERT with whole-word masking is effective in classifying the data accurately (Table 4).

Due to limited time and computational resources, we were unable to perform extensive hyperparameter tuning for each model and in the case of Chinese, only the ChineseBERT-WWM language model was explored. As a result, we had to use the same hyperparameters across languages. Given more time and computational resources, a more thorough hyperparameter tuning process would be preferred so as to ensure that each model was optimized to its fullest potential. In addition, other language models such as MacBERT (Cui et al., 2021) would have been explored as well.

5.3 Limitations

Sarcasm

Sarcasm plays a large role in the way humans communicate, it is no different when it comes to text communication. The presence of sarcasm complicates the meaning behind what was communicated. In the case of this project, sarcastic comments make its emotion or stance at face value differ from its underlying emotion or stance. As such, while deep learning models can recognize patterns to make predictions, they may not be able to detect sarcasm due to its reliance on context, tone of voice, and irony. Even for humans, detecting sarcasm in comments can be difficult when the context is ambiguous or the tone is unclear.

To account for sarcastic comments, when manually labeling the data, we identify cases of sarcasm to the best of our abilities and label them with their underlying emotions and stances. By correctly labeling these texts with their underlying emotion and stance instead of their surface-level emotion and stance, we provide the model with more "accurate" data to learn from. Ultimately, by feeding more labeled sarcastic texts into the model, the model will be able to learn better and make better predictions.

Irrelevant Comments

Irrelevant comments such as spam or advertisement not only do not contribute to the underlying emotion or stance of the text but add noise to the dataset. This may mislead the model into learning patterns that do not accurately represent the target labels. Moreover, such comments are irrelevant to the project scope. To ensure that the model focuses only on learning patterns related to the target emotions and stances, we manually review the dataset and remove these comments. As this is not a scalable solution, it is advisable to explore the spam filtering method before feeding the data into the model.

Subjectiveness of Manual Labeling

Subjectivity in the data labeling process is more prone to occur when there is no ground truth for the label. This is especially so for 7 psychological emotions where there is overlap across certain labels such as 'anger' and 'disgust', as illustrated by the example:

“Wah this place so dirty, how can the govt allow this to happen??”

This subjectivity in interpretation and thus, the data labeling process introduces biases in the labeled data, which inevitably introduces biases in the model training process. This may result in the model either being biased toward certain emotions or hindering the model from reaching the global minima. To mitigate this issue, multiple individuals would label each data point and the majority label will be used as the final label.

6 Integration

The other key aspect of this project involves close collaboration with the other team that is focusing on explainable AI. We had to build a model that enabled the ease of integration with the other team's explainable AI framework. To ensure that we meet our project goals, we have supplemented our sister team with the following deliverables:

	Deliverable	Remarks
1	Train and test datasets used for English and Chinese models	
2	Codes for English and Chinese models	Consists of the following additional functions requested by the other team: 1. process_data Takes in a list of [text, entity] and returns a list of the processed text. The text length will be less than or equal to 50 and the entity within the text has been replaced with the entity token. 2. predict Takes in a list of text and further processes it by constructing the auxiliary sentence. The processed text is then tokenized before feeding to the model and returning the model predicted probabilities for emotions and stances.
3	Model weights for English and Chinese models	

Table 5: Deliverables

These deliverables facilitate the smooth integration of the respective components into the explainable AI framework.

7 Web Application

To provide a more extensive and user-friendly solution, our team went beyond the initial scope of creating a notebook widget, by developing a web application. This intuitive application enables users to effortlessly upload datasets in their preferred language (English or Chinese), showcase the dataset's descriptive statistics, perform data filtering, and obtain the labeled dataset for further analysis. The web application was developed using Streamlit, a powerful framework for creating and deploying data science applications.

After the user uploads their dataset in the correct format, the web application will call upon the backend components to label the dataset by leveraging the deep learning model. After which, the descriptive statistics of the labeled data are then displayed on the web application. These statistics include the overall breakdown of labeled data by emotion and stance, as well as the breakdown of labeled data by emotion and stance for specific entities of interest selected through a filter function. This functionality allows users to gain valuable insights into the emotions and stances present in their dataset, enabling them to better understand their data and make informed decisions about their research. Finally, a file download function is also present for users to download the labeled data in Excel format, allowing them to perform additional analysis as needed.

Overall, our web application provides a streamlined and user-friendly interface for processing, labeling, and analyzing unlabeled textual datasets. By leveraging the power of Streamlit, we have been able to create a powerful and flexible tool that can be customized to meet the specific needs of our end users. Figure 5 shows an example of the web application interface, including the file upload and model language functions. Figure 6 demonstrates the filter function, which enables users to select specific entities of interest and view the percentage breakdown of labeled data by emotion and stance for those entities.

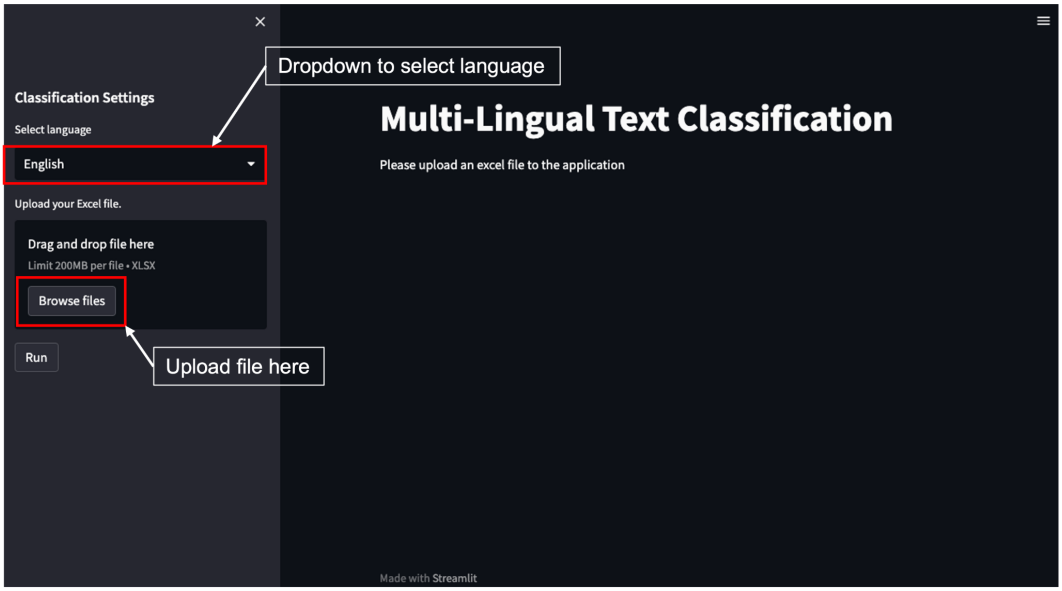


Figure 5: Landing Page of Web Application

Figure 6: Filtering function of Web Application

8 Training Script

Over time, the performance of machine learning models can deteriorate, which is often referred to as model drift. This degradation is mainly caused by the change in the data distribution between the train and production datasets. To ensure the continued relevance and performance of the model, it is necessary to retrain it periodically. To aid in this process, we have made available a configurable training script on GitHub. With just a single command, the training script can retrain the model end-to-end. To further enhance the flexibility of the training script, we have included command shell arguments that allow users to customize the data split, model architecture, and model training process to their specific needs. This enables users to keep their machine learning models up-to-date and maintain their high level of accuracy.

9 Use Case

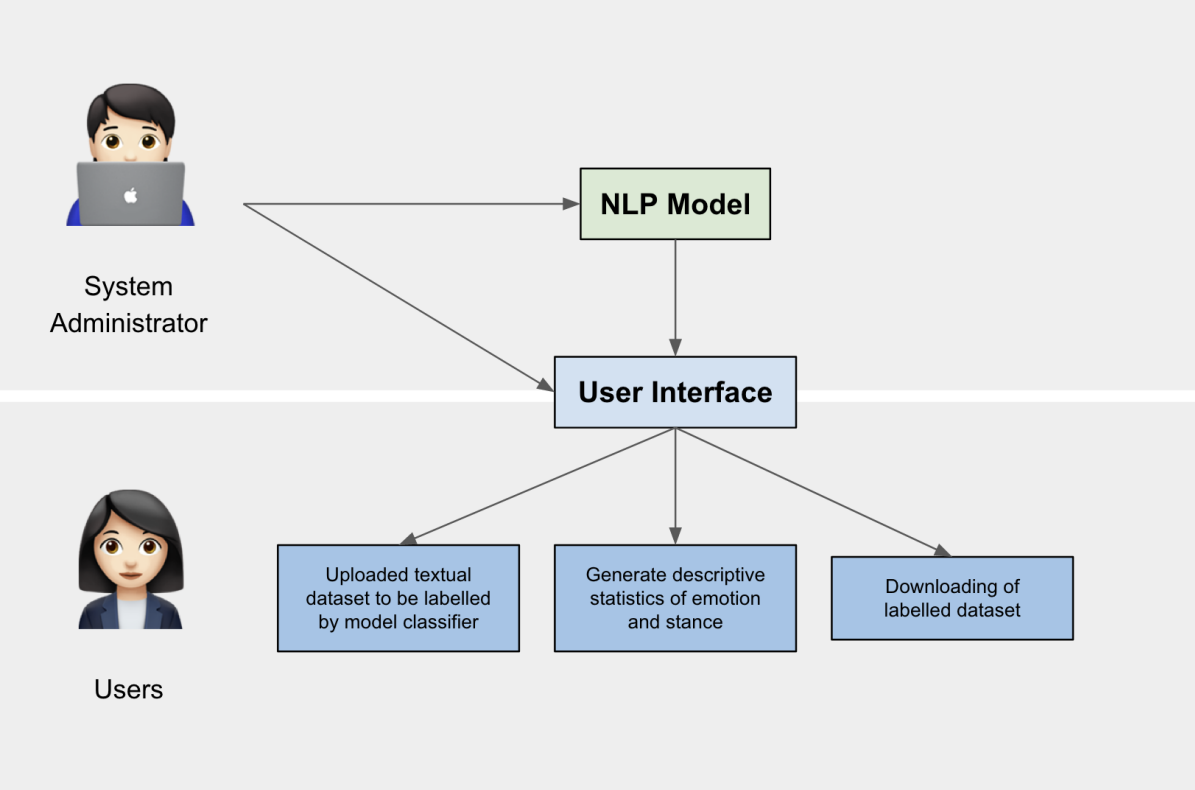


Figure 7: Use Case Diagram

The system administrator can easily retrain the model end-to-end with a single command using the training script provided and their dataset of choice. This not only ensures the model remains relevant but enables the model to adapt to other domains thus, enhancing its accuracy and relevance to the user. The model can then be seamlessly integrated into the user interface, which is maintained by the system administrator as well.

To facilitate the classification process, the user interface allows users to easily upload their unlabelled textual datasets. The interface is designed with simplicity and ease of use in mind, providing clear instructions and a straightforward uploading process. To ensure that users are able to effectively utilize the system, a comprehensive user interface, and model documentation are provided. Once the model classifier has labeled the data based on emotions and stances, descriptive statistics of the labeled outputs will be generated. Additionally, users have the option to download the labeled datasets for further analysis.

Overall, the user interface serves as a user-friendly solution for automated data labeling without requiring any coding knowledge. This alleviates the need for laborious manual data labeling efforts and streamlines the data processing workflow, making it an essential tool for data analysis for research purposes.

10 Recommendations

This section aims to provide recommendations for improving the performance of the models developed to better suit the use case. The recommendations are based on the feedback from stakeholders, as well as areas we felt are worth exploring.

10.1 Spam Detection

We recommend incorporating topic classification and spam detection models as a three-stage pipeline to improve the accuracy and reliability of the emotion and stance prediction models. This approach filters out social media comments that are irrelevant to the topic of discussion, reducing noise in the dataset that is used for training the model, and potentially results in a more reliable model. There are several Python based topic classification libraries, such as *GenSim* and *BERTopic*, that can be used to analyze the social media post and the respective comments. As for spam detection, *SpamPy* is a spam filtering module that can be trained for this context.

Besides reducing noise found in the dataset, picking up spam comments can also provide value to government organizations by identifying scams that are prevalent in social media. These social media scams pose a serious threat to society thus, having a scalable solution is crucial in the fight against scams.

10.2 Incorporating Non-Textual Data

Non-textual digital media has become an integral part of online communication and is often used in conjunction with the text to convey the user's emotions and enhance engagement of the message. Emojis and GIFs have proven popular amongst netizens, with a study showing that 92% of online consumers use Emojis when expressing themselves online (Emogi Research Team, 2016). Hence, these visual cues can bring about great value in the prediction of an entity's emotion and stance.

The integration of Emojis into prediction models has been well-researched, with various Python libraries available for use. For instance, Demoji is able to replace Emojis in a text input with their textual descriptions (i.e,  will be replaced by :shortcake:). In addition, the library *Deepmoji* can be applied in cases where Emojis are used for more complicated contexts of irony and sarcasm. *Deepmoji* has been trained with 1.2 billion tweets containing Emojis and is able to predict the emotional content of a given text based on the presence and position of emojis within it.

Unlike Emojis, GIFs are relatively more complicated in the application as they are short looping clips. As such, image encoding techniques may have to be implemented to extract the emotions and stances expressed. We have received feedback that users would like to incorporate GIFs into the analysis as they are commonly used by netizens. However, incorporating GIFs into a NLP task is still an emerging area of research and there are limitations in the computational resources required, making it a challenging task.

10.3 Mapping Singapore Chinese to Mainland Chinese

Despite not encountering any instances of unique Singaporean Chinese phrases in the limited dataset used, it is important to consider the potential challenges that may arise in larger datasets. It is possible that Singaporean Chinese phrases could be present in a larger dataset, and these phrases may need to be mapped to their Standard Chinese counterparts in order to improve the accuracy of Chinese language models. For instance, the Singaporean Chinese term for "sweater" is represented as "冷衣", while the Standard Chinese term is "毛衣". By mapping these unique phrases, we can ensure that the model is able to correctly identify and classify them, improving its overall performance in understanding and processing the unique Singaporean context.

11 Conclusion

In summary, our project on Multi-lingual Text Classification aimed to develop a system that leverages deep learning models to automate the identification of emotions or stances from texts in different languages, namely English and Chinese. Through this project, we aimed to address the existing challenges faced by our host organization in the manual labeling of data, which is slow, costly, and unscalable.

We developed a robust double-headed entity-based sentiment classifier that utilizes pre-trained models such as BERT and ChineseBert-WWM. These pre-trained models are trained on a large language corpus, and as a result, our system is applicable across a wide range of topics for our host organization;’s research purposes.

Furthermore, we developed a user-friendly web application that allows users to upload their unlabeled textual datasets and obtain labeled datasets for further analysis. Our web application features data validation, descriptive statistics, and a file download function that enable users to gain valuable insights into their data.

Overall, our project has successfully achieved its objectives and provides a practical solution for our host organization to improve the efficiency and accuracy of their research process. The system and web application developed through this project can also be applied to other research areas that require the classification of emotions or stances from textual data.

12 References

Baldini Soares, L., FitzGerald, N., Ling, J., & Kwiatkowski, T. (2019). Matching the Blanks: Distributional Similarity for Relation Learning. https://doi.org/10.48550/arXiv.1906.03158

Cui, Y., Che, W., Liu, T., Qin, B., & Yang, Z. (2021). Pre-Training with Whole Word Masking for Chinese BERT. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *29*, 3504-3514. https://ieeexplore.ieee.org/document/9599397

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. https://doi.org/10.48550/arXiv.1810.04805

Driess, D., Xia, F., S. M. Sajjadi, M., Lynch, C., Chowdhery, A., Ichter, B., Wahid, A., Tompson, J., Vuong, Q., Yu, T., Huang, W., Chebotar, Y., Sermanet, P., Duckworth, D., Levine, S., & Va, V. (2023). PaLM-E: An Embodied Multimodal Language Model. https://doi.org/10.48550/arXiv.2303.03378

Emogi Research Team. (2016). *2015 Emoji Report*. 2015 Emoji Report. Retrieved April 2, 2023, from https://cdn.emogi.com/docs/reports/2015_emoji_report.pdf

E. Peters, M., Neumann, M., Lyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018). Deep contextualized word representations. https://doi.org/10.48550/arXiv.1802.05365

Gupta, S., & Gupta, A. (2019). Dealing with Noise Problem in Machine Learning Data-sets: A Systematic Review. *Procedia Computer Science*, *161*, 466-474. https://doi.org/10.1016/j.procs.2019.11.146

Jiang, L., Yu, M., Zhou, M., Liu, X., & Zhao, T. (2011). Target-dependent Twitter Sentiment Classification. *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, 151-160. https://aclanthology.org/P11-1016

Lin, T.-Y., Goyal, P., Girshick, R., He, K., & Dollár, P. (2017). Focal Loss for Dense Object Detection. https://doi.org/10.48550/arXiv.1708.02002

Loshchilov, I., & Hutter, F. (2017). Decoupled Weight Decay Regularization. https://doi.org/10.48550/arXiv.1711.05101

13 Appendices

13.1 User Guide

13.1.1 Installation Guide

Prerequisites

- Windows/Linux/Mac machine running Python3.8
- Latest version of pip, git, and virtualenv or Anaconda installed

Steps

1. Open the Command Prompt (or Terminal for Mac users) and navigate to your preferred directory.

```
> cd <specified_path>
```

2. Clone the project repository by typing the following command into the terminal.

```
> git clone https://github.com/chengjiangg/BT4103-Group3.git
```

3. Navigate to the user_interface folder using the following command.

```
> cd BT4103-Group3/user_interface
```

4. Create and activate the virtual environment using the following commands.

a. Using Anaconda

```
> conda create --name env
> conda activate env
```

b. Using virtualenv

```
> virtualenv env
> source env/bin/activate
```

OR

```
> virtualenv env
> env\Scripts\activate
```

5. Install the necessary packages using the following command.

```
> pip install -r requirements.txt
```

6. In the folder ‘user_interface’, create a folder ‘saved_models’ and add the following files.
    a. en_model_weight.pth
    b. zh_model_weight.pth

13.1.2 Web Application Guide

Steps

1. Open the Command Prompt (or Terminal for Mac users) and navigate to the cloned directory.
2. Navigate to the user_interface folder using the following command.

```
> cd user_interface
```

3. Activate the virtual environment using the following commands.

a. Using Anaconda

```
> conda activate env
```

b. Using virtualenv

```
> source env/bin/activate
```

OR

```
> env\Scripts\activate
```

4. Start the application using the following command.

```
> streamlit run MyApp.py
```

5. Wait for the web application’s landing page to load in your active web browser.

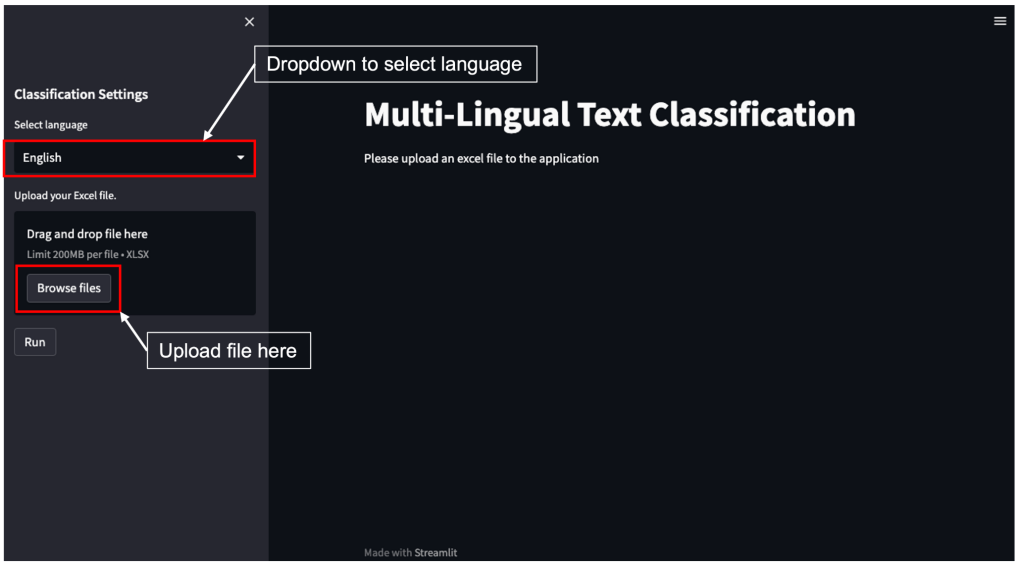


Figure 8: Web Application in Active Web Browser

4. On the sidebar, locate the "Classification Settings" section and select the language (English or Chinese) applicable to your dataset.

5. Click on the "Browse files" button, and your computer's file explorer should appear.

    a. Ensure that your dataset file meets the following requirements:

        i. It should be an Excel file (.xlsx) with a maximum file size of 200 MB

        ii. It should contain at least two rows of data with the first row being the column headers “text” and “entity” (without capitalization)

        iii. It should have two columns named "text" and "entity".

	text	entity
2	eh, always talk about politics, i dun even understand all the jargon and terms they use need to go and use google for everything waste my time	politics
3	I think we need more young blood in politics, you know can bring in fresh perspectives and ideas	politics
4	eh, but have you all noticed that pap and wp just blame each other for everything, instead of taking responsibility like they should ah	pap
5	eh, but have you all noticed that pap and wp just blame each other for everything, instead of taking responsibility like they should ah	wp
6	always saying china and russia committing this and that crime against humanity true lah they did all that but what about the us and europe ah they also didn acknowledge wat	china
7	always saying china and russia committing this and that crime against humanity true lah they did all that but what about the us and europe ah they also didn acknowledge wat	russia

Figure 9: Required Excel Format

b. The following error message will appear if the excel is in the incorrect format

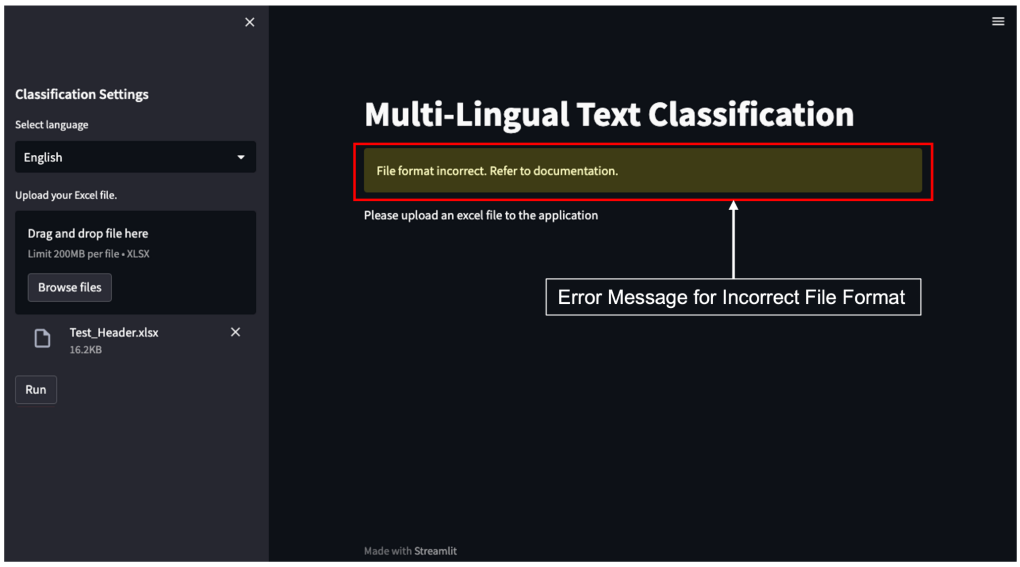


Figure 10: Error Message for Incorrect File Format

6. Navigate to the location where you saved your dataset file and select it.

7. Click on “Run”. The model will begin labeling your dataset. The time taken depends on the number of data points and the computer. An approximation of the duration would be 10 seconds for every 60 rows.

8. Once the model has finished labeling the datasets, the dashboard will display the following descriptive statistics of the labeled data.

    a. A breakdown of emotions and stances

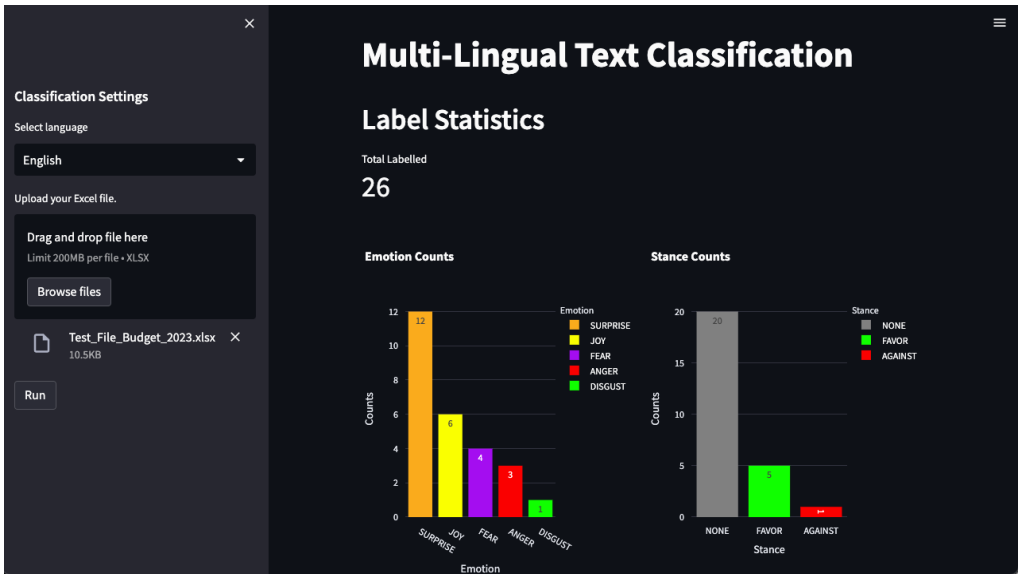


Figure 11: Breakdown of Emotions and Stances

b. An autocomplete field for filtering entities of interest

Figure 12: Autocomplete Field to Filter Entities

c. A table of the labeled data and an option to download the labeled data

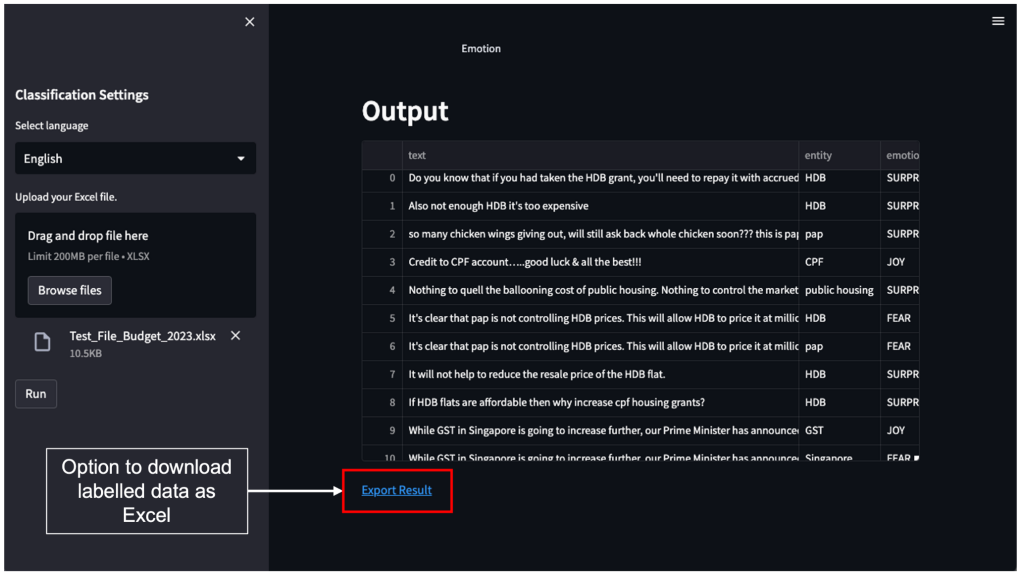


Figure 13: Table of the Labeled Data

9. Click on the "Export Result" button to download the labeled dataset as an Excel file.

10. Ctrl + C in the terminal to shut down the application

11. Type the following commands in the terminal to deactivate the virtual environment

```
> deactivate
```

13.1.3 Retraining Guide

Steps

1. Compile the new datasets and add them to the ‘data’ folder under the ‘training_script’ folder.
2. Open the Command Prompt (or Terminal for Mac users) and navigate to the cloned directory.
3. Navigate to the training_script folder using the following command.

```
> cd training_script
```

4. Retrain the model using the following commands.

a. For English

```
> python main.py --excel_filename <eng_dataset_name>.xlsx
--sheet_name Sheet1 --classifer_type en
```

b. For Chinese

```
> python main.py --excel_filename <chi_dataset_name>.xlsx
--sheet_name Sheet1 --classifer_type zh
```

5. The following will appear in the terminal upon successful execution of the training script.

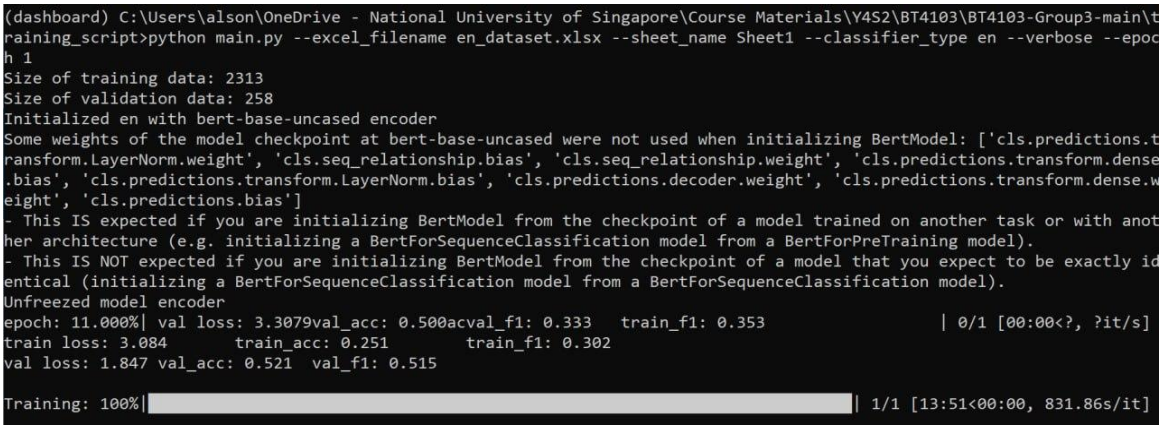


Figure 14: Successful Execution of Training Script

13.2 Sentiment Classification Report Breakdown

The following tables report the in-depth breakdown of the classification results for emotions and stances, for the respective models.

BERT

Emotion	Precision	Recall	F1-Score
Anger	0.90	0.95	0.93
Disgust	0.82	0.82	0.82
Fear	0.89	0.82	0.87
Joy	0.91	0.91	0.91
Sadness	0.73	0.80	0.76
Surprise	0.95	0.95	0.95
Neutral	0.95	0.86	0.90

Table 6: BERT Model Performance (Emotions)

Stance	Precision	Recall	F1-score
Against	0.96	0.93	0.95
Favor	0.83	0.89	0.86
None	0.93	0.93	0.93

Table 7: BERT Model Performance (Stance)

SingBERT

Emotion	Precision	Recall	F1-Score
Anger	0.77	0.50	0.61
Disgust	0.47	0.53	0.50
Fear	0.62	0.80	0.70
Joy	0.78	0.82	0.80
Sadness	0.56	0.45	0.50
Surprise	0.62	0.71	0.67
Neutral	0.57	0.55	0.56

Table 8: SingBERT Model Performance (Emotions)

Stance	Precision	Recall	F1-score
Against	0.73	0.66	0.69
Favor	0.69	0.67	0.68
None	0.67	0.77	0.71

Table 9: SingBERT Model Performance (Stance)

ChineseBERT-WWM

Emotion	Precision	Recall	F1-Score
Anger	0.78	0.78	0.78
Disgust	0.78	0.88	0.82
Fear	0.75	0.75	0.75
Joy	0.67	1.00	0.80
Sadness	0.67	0.75	0.71
Surprise	0.75	1.00	0.86
Neutral	0.80	0.40	0.53

Table 10: ChineseBERT Model Performance (Emotions)

Stance	Precision	Recall	F1-score
Against	0.86	1.00	0.92
Favor	0.75	1.00	0.86
None	1.00	0.77	0.87

Table 11: ChineseBERT Model Performance (Stance)

13.3 Label Sentiment Heatmap

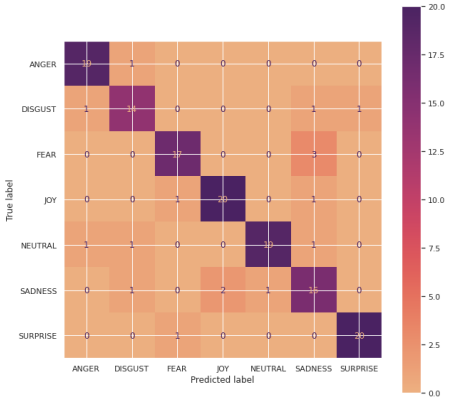
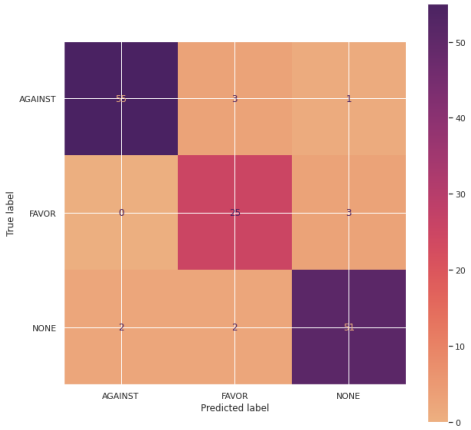
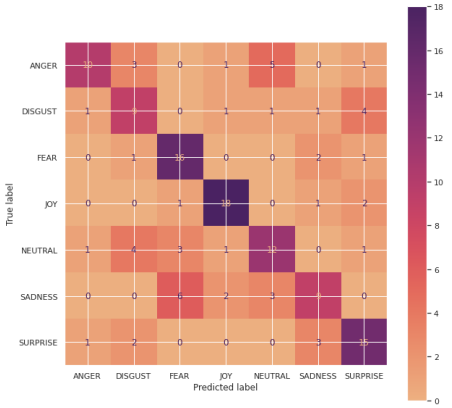
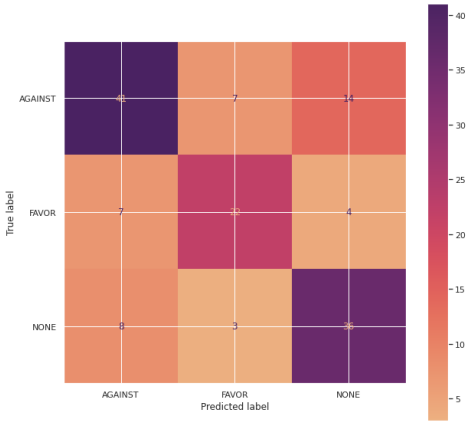
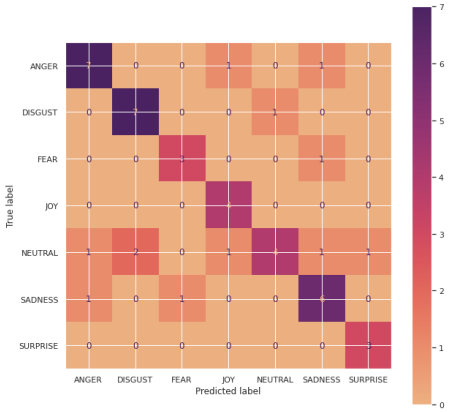
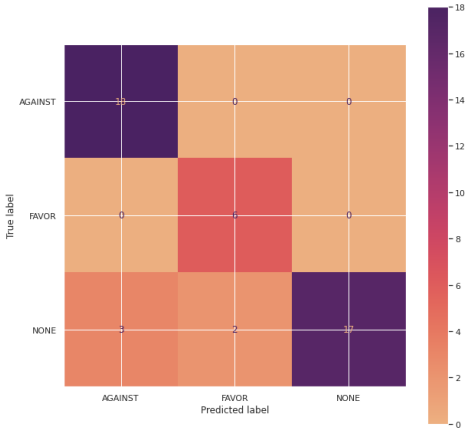
Model	Emotions	Stances
BERT		
SingBert		
ChineseBERT-WWM		

Table 12: Label Sentiment Heatmap