NANYANG TECHNOLOGICAL UNIVERSITY
SINGAPORE

IE4483: Artificial Intelligence and Data Mining

Project Name: Sentiment Analysis (Option 1)

School of Electrical and Electronic Engineering

Academic Year 2023/24
Semester 1

Group Members:

Chen Hung-Jui (U2020125F)
Ivan Tan Kah Keng (U2123310H)
Wong Yu Hao (U2120641H)

Content Page

Choice of Feature Format

Subword Tokenization

Tokenization is a popular technique used in Natural Language Processing (NLP) where sentences are sliced into smaller units for easy understanding of machines [1]. The task of tokenization may look easy at first glance, but symbols or punctuations like “@”, “%”, or even “ ’ ” will greatly affect the results of NLP since they alter the meaning of the whole sentence.

Tokenization can be done by several approaches such as Word Tokenization, Character Tokenization and Subword Tokenization [1]. The approach that we are using in our project is the Subword Tokenization [2], as such an approach effectively solves the problem of Out Of Vocabulary (OOV). During tokenization, words that are unknown to the machine (i.e., words that are not trained with) are replaced by “unknown” tokens which affect the accuracy of NLP [1]. Such tokens are significantly reduced using Subword Tokenization and valuable inferences can be made by the machine on how a word functions in a sentence [1]. For example, the word “eating”, which is OOV, will be separated into two tokens “eat” and “ing”. A machine that only understands the meaning of “eat” will be able to interpret the meaning of “eating” by getting hints from the “ing” suffix.

Data Preprocessing

The JSON files are read as pandas DataFrame. The “reviews” column is then cast to the “string” data type as it is wrongly identified by pandas as an “object” data type.

We tokenize our data by utilising the HuggingFace Tokenizer. HuggingFace is a platform known for its rich library of Transformer and NLP models, which also allows users to freely upload their trained and fine-tuned models, much like GitHub. The HuggingFace Tokenizer leverages different algorithms to tokenize the data based on the model selected by the user, namely Byte-Pair Encoding (BPE), WordPiece and Unigram [2]. The algorithm that we are using for our project is the WordPiece algorithm [2].

The WordPiece algorithm first creates a base vocabulary including all characters and their frequencies from the set of unique words found in the data [2]. Then, it learns rules that merge two characters together to form new sets of vocabulary until the desired vocabulary size is reached [2]. The rules are determined by the following formula [3]:

$$score = \frac{frequency\ of\ pair}{frequency\ of\ first\ element * frequency\ of\ second\ element}$$

Pairs with the highest score will be merged and eventually the vocabulary will be finalised [2]. This vocabulary is then used to split words into subword tokens, as shown in the “eating” example above.

Model Selection

After careful consideration of the task nature, time constraints and limited hardware resources, our group decided to use a pretrained Large Language Model (LLM) available in the open-source community as a base model. The model is then further fine-tuned to learn from our dataset so that it gives a more accurate classification in our specific task.

DistilBERT, which is a distilled and lightweight version of the state-of-the-art Bidirectional Encoder Representations from Transformers (BERT) model, is an excellent candidate for our situation, as it provides us with a 40% smaller and 60% faster solution while preserving 97% of its original counterpart's performance [4].

Model	Macro-score	CoLA	MNLI	MRPC	QNLI	QQP	RTE	SST-2	STS-B	WNLI
BERT-base-uncased	79.5	56.3	84.7	88.6	91.8	89.6	69.3	92.7	89.0	53.5
DistilBERT-base-uncased	77.0	51.3	82.1	87.5	89.2	88.5	59.9	91.3	86.9	56.3

Table 1. Score comparisons between BERT and DistilBERT measured in the General Language Understanding Evaluation (GLUE) benchmark [4]

As DistilBERT was pretrained using the original BERT as a teacher model [4], it inherited the characteristics of BERT and is suitable for NLP tasks such as summarization, question answering, polysemy resolution and sentiment analysis [5]. Thus, DistilBERT is chosen as our base model.

Model Architecture

Since its introduction by Google in 2018 [5], BERT has become a widely favoured model in executing NLP tasks. As suggested by its name, BERT attempts to learn the meaning of a word in a bidirectional manner [5], i.e., by considering words come before and after the current word. This is only made possible by utilising the Transformer architecture [5].

By leveraging its self-attention mechanism, Transformers are able to learn the relationship between each input element and decide which elements are important [6], or in other words, are worth "attention". When a sentence is passed into a Transformer, these "attention" weights will help in deducing the context of the sentence [6]. Transformers are also well suited for unsupervised learning with a huge amount of

data points, as computational resources are not wasted on those deemed not worth "attention" [6].

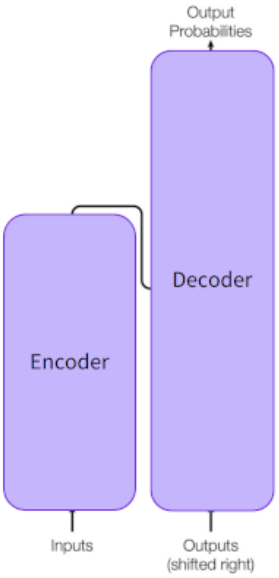

Figure 1. Simplified diagram of Transformer architecture [6]

BERT was pretrained on the unlabelled Wikipedia and BookCorpus datasets, consisting of more than 3.2 billion words, using an encoder-only Transformer architecture [6]. The texts were tokenized by applying the WordPiece algorithm, to form a vocabulary of size 30,000 [7]. These tokens were then summed with segmentation embeddings and position embeddings, before being trained to complete two tasks, namely Masked Language Model (MLM) and Next Sentence Prediction (NSP) [7].

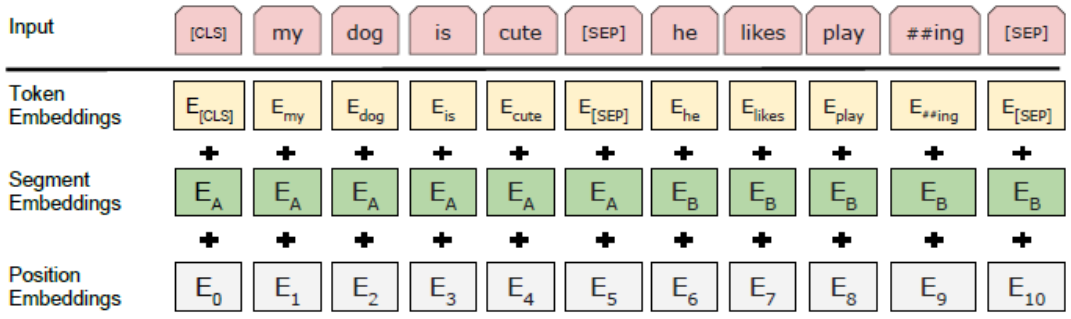

Figure 2. BERT input representation [7, Fig. 2]

In the MLM training, 15% of the tokens were selected as masked tokens [7]. However, only 80% of the masked tokens were eventually masked, while half of the remaining tokens were replaced by random different tokens, and the other half remained unchanged [7]. BERT was to predict the masked tokens by learning bidirectionally from the other tokens.

On the other hand, the NSP training required BERT to predict if a sentence is the subsequent sentence of another sentence. 50% of the input pairs were true subsequent sentence pairs, while the other 50% were not [7]. The MLM and NSP training were executed concurrently.

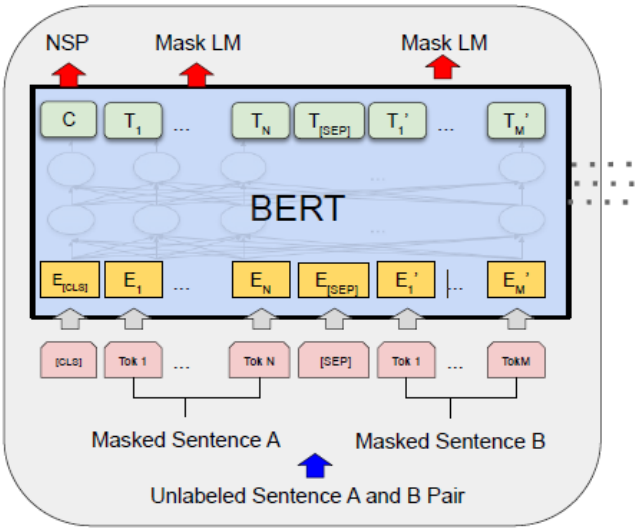

Figure 3. Overview of BERT pre-training [7, Fig. 1]

As a rule of thumb, the pretrained BERT will be further fine-tuned with labelled data to perform better in specific tasks [5]. Nevertheless, this round of supervised learning is not as expensive as the pre-training. The pretrained BERT is open-sourced and can be fine-tuned by anyone with relatively smaller datasets and a shorter training time [5]. For sentiment analysis, a classification layer is added on top of the Transformer output [8].

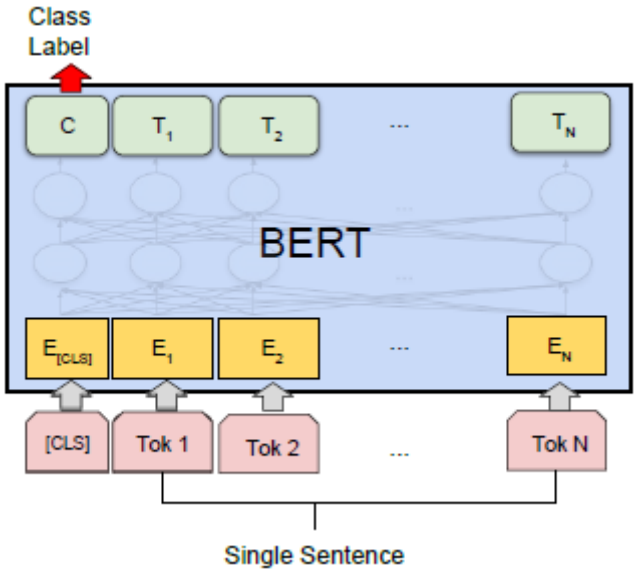

Figure 4. Overview of BERT fine-tuning [7, Fig. 4(b)]

DistilBERT was pretrained on the same datasets using the knowledge distillation technique and shares a similar architecture with its teacher model, BERT [4]. DistilBERT has a total of 6 layers, 768 dimensions, 12 heads and 66 million parameters [4]. The input size of DistilBERT is limited to not more than 512 tokens [9] and anything more than that will need to be trained by batches.

Loss Function

The loss function used during our fine-tuning is the cross-entropy loss. It is a commonly used metric for classification tasks, and it penalises greatly for predictions that are confident but wrong [10]. It measures the performance of a classification model, where a lower value indicates a better performance [10]. It is computed using the following formula [10]:

$$loss = -(y\,log(p) + (1-y)\,log(1-p))$$

where y = 0 or 1 depending on whether the label is predicted correctly, and p = predicted probability that a sample is of the predicted label (i.e., a model's confidence in its output).

Training Strategy

The data in "train.json" is split into three portions, namely the training data, validation data, and test data. The split is done by using the scikit-learn's train_test_split with a fixed random seed to ensure a consistent result. It is also stratified to maintain the same proportion of different sentiments in each portion. The percentages of each portion are as follows:

Training data	70%
Validation data	15%
Test data	15%

Table 2. Percentages of each portion for train test split

The training data are passed into the model for training (i.e., fine-tuning of the pretrained DistilBERT), while validation data are used to prevent overfitting. We leveraged the early stopping technique so that the training is automatically stopped if the validation loss does not decrease for 3 epochs. The model weights are restored to the epoch with minimum loss afterwards.

The test data are used to compare the performance of the model pre-training and post-training. The metrics we measured are the cross-entropy loss, accuracy, and the Area Under the Curve of the Receiver Operating Characteristic curve (AUC ROC). Decrease in cross-entropy loss and increases in accuracy as well as AUC ROC indicate a successful training.

How to run the code

The Jupyter Notebook used is submitted together with this report (File name = *"sentiment_analysis.ipynb"*). Alternatively, you can access the code via Google Colab: https://colab.research.google.com/drive/1IXtTGuPSR0gYUQMRjJiQc2x-W2ZI7ZIM?usp=sharing

1. Ensure "train.json" and "test.json" are in the same directory as the Notebook / uploaded to Google Colab.
2. Ensure relevant packages are installed in your environment.
3. Restart and run all the code.

The structure of the code is as follows:

1. Installing and Importing of Packages
2. Reading and Wrangling of Data
3. Train Test Split
4. Pre-training Performance (Before Fine-tuning)
5. Training of Model (Fine-tuning)
6. Post-training Performance (After Fine-tuning)
7. Prediction of Sentiments in "test.json"
8. Export the Results to "submission.csv"

Note that the results are slightly different for every run probably due to Dropout being used in BERT [11]. We have not found a way to solve this issue yet.

Model Parameter

Learning Rate

The learning rate is a crucial model parameter in the training of machine learning models. It determines the size of the step at each loop while moving toward a minimum of a loss function.

At the low learning rate the network slowly converges and the optimization process progresses slowly, making the model take smaller step size at each loop.

However, the high learning rate will cause the network to take larger step sizes during training which can lead to overshooting the minimum of the loss function.

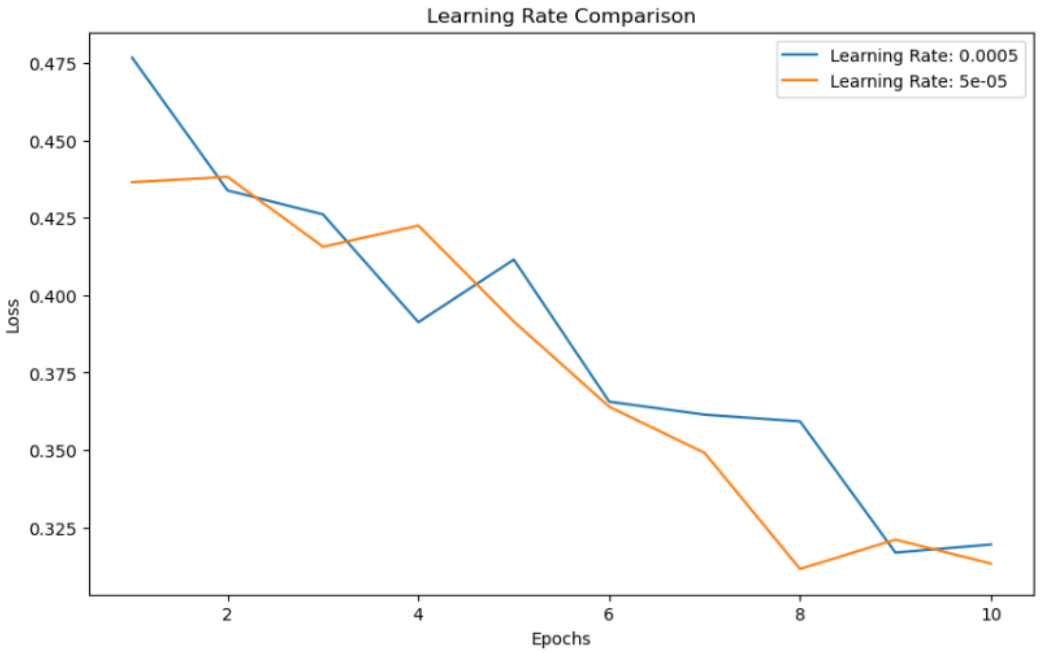

Figure 5. Loss vs Epochs comparison on different learning rates

Based on this observation above, both of the learning rates result in a decreasing loss over epochs, indicating that the model is learning and improving its performance. The plot of this graph describes the choice of learning rate that affects the convergence speed during the training process. Therefore, our team chose to set our learning rate lower to prevent our model from overfitting. The specific value of 5e-5 is chosen because it is one of the recommended learning rates in the original BERT paper [7].

Batch Size

Batch size is a hyperparameter that determines the number of the training examples being processed before each update of the model's weights. The number of batches has implications for both computational efficiency and the quality of the model.

- Computational Efficiency: Efficiency the available computing resources are utilised to perform the necessary computation during the training phase. It involves consideration of both time efficiency and memory efficiency.
    - Time efficiency
        - How quickly the model can be trained
    - Memory efficiency
        - How effectively the available memory is used during training

- Quality of the model: Influence how well the trained model generalizes to unseen data.

The advantages and disadvantages of different batch sizes are as follows:

Large batch size

- Advantages
    - Computational Efficiency: Making use of the parallelism available in modern hardware and facilitating more efficient processing
    - Memory Stability: Optimizing GPU memory usage which contribute to effective memory utilization

- Disadvantages
    - Memory Requirements: The need for more GPU memory can pose limitation, particularly on devices with constrained memory capacity

Smaller batch size

- Advantages
    - Memory Efficiency: Demanding less GPU memory, smaller batch sizes are proved to be resource efficient.

- Disadvantages
    - Computational Inefficiency: The underutilization of GPU parallel processing capabilities in smaller batch sizes may result in prolonged training times.

Figure 6. Test error rate and time taken comparisons on different batch sizes [1]

Based on the obeservation from the graphs, the batch size of 32 has the lowest error rate compared to the other batch sizes. In this scenario, opting for a smaller batch size is driven by the constraint of limited GPU memory available for training. Therefore, our team has chosen the suitable training batch size of 16, which is also recommended in the original BERT paper [7].

Number of Epochs

The number of epochs is the number of the iteration to complete passes through the training dataset. During each epoch of training, the model is exposed to a subset of the training data.

As we are using the early stopping technique, the training of the model has been automatically stopped after the 5th epoch. Weights are then restored to the 2nd epoch, which has a minimum loss. Opting for a lower loss helps prevent overfitting of the training data.

Model Performance

An overview of our model's performance is as follows:

	Loss	Accuracy	ROC AUC
Pre-training (Before fine-tune)	0.6592	0.8434	0.9280
Post-training (After fine-tune)	0.1598	0.9388	0.9628

Table 3. Model's performance before and after training

	Predicted Negative (0)	Predicted Positive (1)
Actual Negative (0)	142	21
Actual Positive (1)	153	795

Table 4. Confusion Matrix (Pre-training)

	Predicted Negative (0)	Predicted Positive (1)
Actual Negative (0)	112	51
Actual Positive (1)	17	931

Table 5. Confusion Matrix (Post-training)

From the measured metrics, we conclude that the model performs better after training as its loss is greatly reduced.

Correctly and Incorrectly classified samples in Test set

The predicted sentiments of “test.json” can be found in the “submission.csv”.

There are 4 different scenarios that our model will produce in the test set, such as true positive (TP), true negative (TN), false positive (FP) and false negative (FN).

Classification

True Positive (TP): An outcome where the model correctly predicts the positive class

True Negative (TN): An outcome where the model correctly predicts the negative class

False Positive (FP): An outcome where the model incorrectly predicts the negative class as positive

False Negative(FN): An outcome where the model incorrectly predicts the positive class as negative

1. Classified the **positive** sentiments **correctly [TP]**

Reviews	Predicted Sentiments
I am a deployed Navy Reservist. I am currently assigned to a tactical expeditionary unit deployed to the middle-east where I serve in a TACCOMM/Tactical Crypto Support role in land and maritime operations. Before deploying, I spent a great deal of time and effort in selecting the 'right boot' in which I knew I would be spending a great deal of time. I tried Oakley, Converse, 5.11 and of course the Bates M-9. I felt that there was no contest. The Bates was by far the most **comfortable** and seemed to have the most **functional** sole design and least potentially irritating upper, and so on. Let me re-wind a little bit. In my civilian life, outside of the Reserve (which at this point seems a distant memory) I am an avid runner, to the tune of about 70 miles per week. While I've run many hundred of miles on roads, my focus over the past couple of years has shifted to off-road 'trail runs' and ultra-marathon courses. I know my feet. I know what works in a shoe that is going to be demanded to perform in horrible conditions. I have had great success over time with literally one pair of shoes from one manufacturer and as such I bought many pairs of that shoe. In June of 2006, I bought my first pair of M-9 Desert Assault boots to take on my July deployment. I decided	1

that in order to break them in, I'd take them on the trail. I figured that they'd tear my feet up a bit, but it wouldn't be the first time a new shoe had done so. The day I first wore them I set off with an 85lb. pack into a tick infested trail in a local state park. It was about 100 degrees outside with humidity nearly to match. After 12 miles on the trail I returned home to find that my feet were all but un-fouled! I was **amazed**. I would repeat this process a few times and very soon after ordered two additional pairs of M-9s. Fast forward just a bit. While in pre-deployment training in Coronado, CA, I took to wearing my boots for all of our PT. I also logged about 30 to 50 miles per week of independent running, again, in my M-9s. Most of these runs included a pack with about 50 - 70lbs in it, though now I was running almost entirely on pavement. My M-9's performed without flaw. I left Coronado for the middle east without any doubt that I had the best boots I could have possibly chosen, and they were well broken in to boot. ;) Fast forward some more. Recently I was able to take a two day MWR trip in order to run the Dubai (UAE) Marathon...in my M-9s. While we (there were five of us from my unit that ran the marathon) have had almost no opportunity to do any training runs while in theater, I knew that if I were to tackle 26.2 miles on pavement (man, I hate running on pavement), there was no way I was going to do so without wearing my 'old faithfuls'. They did not **disappoint**! While I took a very conservative pace, my M-9s carried me with ease to the finish line. My Chief, also the proud owner of a well beaten pair of M-9's ran the marathon with me, his first ever. He too wore his M-9s, as his running shoes were simply shot. He was nothing short of elated to have finished his first marathon attempt, and others were nothing short of amazed that he did it in 'combat boots'. If they only knew. ;) I must say, I've never had so many people come up to me during a marathon to chat and in every case, the topic was my choice of footwear. In every instance I would simply tell them with a smile, "You don't understand, these are really **great** boots." In short, Bates created a nearly **perfect** boot. Are there things I would change? Certainly. But even if Bates never changes a thing about them, I will keep buying them until I am not longer able to run, which I assure you will not be for a very long time. I have already scheduled additional marathons and ultra-marathons (off road of course) for after I return home, and I'll let you guess what my choice of footwear will be. I have found my new 'running shoe'.	
These shoes are **durable**, **comfortable**, and most importantly come in different widths. Most other athletic shoe manufacturers	1

apprarently don't make anything but D width. New Balance does. For those of you **complaining** about width, my guess is that if you size your feet correctly, that problem will go away. I had always worn a 12D width, until some gentleman at a mom and pop store sized my feet correctly for the first time in my life I found out I need a 13B. You may think you know your feet, but you may not. As to comfort, I can't **complain**. As to durability, maybe they are a little stiffer, but if that is what extends the life of these shoes over the others, that is well **worth** it in my book	

The model demonstrates an ability to accurately classify positive sentiments, even when there are negative words in the sentence. For example, in the first review, the term ‘disappoint’ was used, and in the second review, the term ‘complaining’ was present.

Despite these negative expressions, the model correctly categorises both reviews as positive sentiments. This accuracy can be attributed to the prevalence of positive words in the review such as ‘comfortable’, ‘functional’, ‘amazed’, ‘perfect’ in the first review, and ‘durable’, ‘comfortable’, ‘worth’ in the second review.

2. Classified the **negative** sentiments **correctly [TN]**

Reviews	Predicted Sentiments
I received the night gown, tried it on, and actually **hated** it. The built in bra was loose, as was the top and there was just too much fabric up top. The straps are too wide. The rest of the gown was okay, although it too seemed too drapey and long. I sent the gown back 3 weeks ago, now, and have STILL not received a refund. I am very **disappointed** with the whole transaction/product	0
Although very **cute**, Crocs are not good shoes. They are a cheap, plastic shoe that is over priced and NOT **comfortable** at all. I bought a pair for my two year old daughter because of the **ease** to clean them and because they are "non slip". She did slip in them, many times and the **worse** part is that they gave her huge terrible red blisters on the side of her feet. They are the right size but I just think they are not made well. Don't go for the hype. They are a **useless** shoe.	0

The model demonstrates an ability to accurately classify negative sentiments, even when there are positive words in the sentence. For example, in the first review, the term ‘okay’ was used, and in the second review, the term ‘comfortable’, ‘cute’ was present.

Despite these positive expressions, the model correctly categorises both reviews as negative sentiments. This accuracy can be attributed to the prevalence of negative words in the review such as ‘hated’, ‘disappointed’ in the first review, and ‘worse’, ‘useless’ in the second review.

3. Classified the **negative** sentiments **incorrectly [FP]**

Reviews	Predicted Sentiments
OK, I found my fiance online, why not my wedding dress -- sounds easy in comparison, right? Well, after three months of looking (offline, too), I was ready to give up. We're having a very small, relatively casual outdoor May wedding, and I couldn't find anything that really fit my idea of what I want to wear. Then I found Tension Dress (bad name!) on Amazon.	1 [Should be 0]
I bought a pair of the Prima Crocs only wore them for 2 days because the inner lining rubs **painfully** on my heel and the shoes turned my feet black. A friend said Crocs were the most **comfortable** and breathable shoes she's ever worn, so I was shocked with the Primas. They're **painful** and don't breathe. I emailed Crocs to ask if there was a way to stretch them out (because they're unwearable) and here was the reply from their customer service rep: "The Prima is our one style that is made for looks and **comfort**. This style is not for everyone, and if they are not stretching to where you need them with wear, they might not be for you. There are only 5 holes in the top of this very slim fitting shoe, so how much air can get through there in the first place. I personally have no problem with this style, so I believe it depends on the person. If you are **dissatisfied** you can try to take them back to where you bought them and see what they can do. Thank you." Cely Adams Customer Service Representative I typically search for reviews before i purchase a product, if I had this info I wouldn't have bought them online, I'd recommend buying them someplace that is easy to return	1 [Should be 0]

The model demonstrates an inability to accurately classify negative sentiments in some cases.

The first review appears to be negative, highlighting the difficulty in finding a wedding dress after three months of searching, along with the initial frustration expressed in the statement, ‘Well, after three months of looking (offline, too), I was ready to give up” suggest a level of dissatisfaction.

The sentiment shifted towards a more positive tone with the discovery of the ‘Tension Dress’ on Amazon. However, this customer was complaining that the original platform used did not solve their problem but Amazon did. Therefore, we can conclude this review should be a negative sentiment.

The second review is predominantly negative. The customer expresses dissatisfaction with getting a pair of Prima Crocs. The terms ‘comfortable’, ‘comfort’ are present in the first sentence. Nonetheless, the negative sentiment arises from the issues with the cover and lack of response from Crocs. The frustration is evident in the underlined statement, “I wouldn't have bought them online” emphasises the negative sentiment due to the unsatisfactory experience. There are also some prevalence of negative words in the review such as, ‘painfully’ and ‘dissatisfied’. Overall, we can conclude the model demonstrates an inability to accurately classify these negative sentiments.

4. Classified the **positive** sentiments **incorrectly [FN]**

Reviews	**Predicted Sentiments**
Well, I recieved this product as it was one of the only cases made for the Video Ipod when I got mine. As soon as i got it, I put it on.... it was a bit difficult at first, but once in there snug, i have never taken it out. About... 2 days later, my IPod fell out of my jacket pocket and tumbled hard down a flight of CONCRETE stairs. I was sure that it was damaged... but, I opened the case, turned it on, and and was fine.... I was quite **relieved**. That alone justifies the price of the case. I am **proud** to say that it is still in there to this day, and the only thing I could have asked for would have been a somewhat **tighter** fit around the clickwheel, as there is a bit of space for **dirt**/**dust** to get into. That and the whole made for the headphones could be a tad bigger.	0 [Should be 1]
Discovered ISOTONER slippers, a decade or more ago, at Macys. Tried imitations out there but NOT as **comfortable** so the other brands were exiled to the deep dark corner of my closet for closet monsters to use. Only buy ISOTONER now and don't share them with closet monsters. These slippers follow me around the house. I have even driven in them. I am wearing a pair right now. I make room for them in my bags when I travel -- never leave home without them. Arch	0 [Should be 1]

support makes all the difference! The women's large is comparable to European size 39. When machine washed and machine dried, they come out looking like **new**. They do wear out eventually or get misplaced around the house so buy two pairs. Amusing aside: Until his feet grew too large, my teenage son used to steal my slippers (navy blue ISOTONER terry). Had to beg him to give them back. Now his feet are too large and the men's slippers I have bought him, he says, are not as comfortable, alas	

The model demonstrates an inability to accurately classify positive sentiments in some cases.

The first review stated the positive experience with the case, highlighting its durability and protective qualities. The customer mentioned that the iPod survived a fall down concrete stairs without damage, which they find impressive and a justification for the price of the case. The positive words in the review such as ‘relieved’, ‘proud’ in the first review.

However, there are minor drawbacks in the first review, stating the word ‘tighter’, ‘dirt’, ‘dust’ as minor negative words in the first review. Overall in the first review, the tone is emphasising the case’s effectiveness in protecting the iPod, which we can conclude the model demonstrates an inability to accurately classify positive sentiment in this case.

Moreover, the second review expressed in the statement is positive. The customer expressed their satisfaction with ISOTONER slippers, stating that they are the most comfortable slippers they have tried. The positive sentiment can be inferred from the statement such as “These slippers follow me around the house. I have even driven in them.” and “never leave home without them”.

However, there are minor drawbacks in the second review, stating “are not as comfortable” carries a negative sentiment. This suggests the customer shares a humorous story about her teenage son not comfortable with the men’s slippers, not the ISOTONER slippers. Overall in the second review, the tone is highlighting the aspects of the shorts that the person appreciates and portray a strong preference for ISOTONER slippers, which we can conclude the model demonstrates an inability to accurately classify these positive sentiments in this case.

Strength and Weakness of the Model

Strength of the model

- Sentiment prediction:
    - The model demonstrates the ability to predict most of the sentiments accurately, even when users express different sentiments at the beginning of their reviews.

- Tokenization:
    - The use of the BERT model tokenizer involves converting text data into tokens, which are then analysed by the BERT model to capture contextual information and generate more accurate predictions

Weakness of the model

- Tonal Analysis of Customer Reviews:
    - There are minor drawbacks of the model that cannot classify the customer's tone correctly based on the review, stating from **FN**, ""These slippers follow me around the house. I have even driven in them." and "never leave home without them" are the positive sentiment in the second review text that the customer appreciates, which the model wrongly classified as negative.
    - The other example, stating from **FP**, "I wouldn't have bought them online" emphasises the negative sentiment due to the unsatisfactory experience, which the model wrongly classified as positive.

Different choice of feature format

In this project, we are using NLP to enable machine learning algorithms in organizing and comprehending human languages. Tokenization is one of the components among the various elements that contribute to the functioning of NLP.

Different choice of tokenization strategy

A tokenizer is a method for breaking down a piece of text into smaller units known as tokens which converts it into a useful data string.

The initial stage in the NLP process involves collecting the sentence and dividing it into comprehensible words.

Take for an example taken from the underline phrase from **FP** of the second review,

“I wouldn't have bought them online”

In order for this sentence to be understood by a machine, tokenization is applied to break the sentence into individual parts. With tokenization, the outcome will split word by word.

‘I’ , ‘wouldn’t’ , ‘have’ , ‘bought’ , ‘them’ , ‘online’.

Consequently, splitting word by word helps the program to understand each of the words by themselves.

As a result, various techniques are employed to separate and tokenize words, and choices of these methods significantly influence subsequent stages of the NLP process.

1. Word Tokenization
The process of breaking down a text into individual words. It can identify the pauses in spaces in text and separates the content into individual words by utilizing delimiters (i.e., ‘,’ or ‘;’) [1].

Taking for the previous example phrase from **FP** of the first review:

“I wouldn't have bought them online”

Processing to this:

‘I’ , ‘wouldn’t’ , ‘have’ , ‘bought’ , ‘them’ , ‘online’.

However, there are some drawbacks of the contraction understanding of the original meaning. Taking the previous example from the above (“wouldn’t”) may be split into separate tokens, potentially affecting the accuracy of the understanding of the original meaning.

Moreover, Word Tokenization faces challenges when dealing with OOV words. It often resorts to substituting unknown words with a generic token. For instance, if there are three ‘unknowns’ word tokens, they can be the same word or entirely different unknown words, which will lead to an inaccuracy of the understood meaning.

2. Character Tokenization
The process involves breaking down a text into individual characters. Instead of dividing text into words, Character Tokenization completely separates the text into individual characters. This approach enables the tokenization process to preserve information about OOV words, which the capability that Word Tokenization lacks [1].

Taking for the previous example phrase above:

“I wouldn't have bought them online”

Processing to this:

' I ', ' ', ' w ', ' o ', ' u ', ' l ', ' d ', ' n ', "' ", ' t ', ' ', ' h ', ' a ', ' v ', ' e ', ' ', ' b ', ' o ', ' u ', ' g ', ' h ', ' t ', ' ', ' t ', ' h ', ' e ', ' m ', ' ', ' o ', ' n ', ' l ', ' i ', ' n ', ' e '

3. Sub Word Tokenization
The process is similar to Word Tokenization, but it breaks down words into smaller meaningful units and further using specific linguistic rules. It is a key technique they employ involving removing affixes. Since prefixes, suffixes and infixes alter the inherent meaning of words, they also can help programs understand a word’s function [1].

Take for an example taken from the underline phrase from FP of the first review,

“Well, after three months of looking (offline, too)”

With sub word tokenization, the outcome will split into subword pieces.

' Well ', ' , ', ' ', ' af ', ' ter ', ' ', ' th ', ' ree ', ' ', ' mon ', ' ths ', ' ’ , ’ of ' ' look’ , ‘ ing ‘ , ' ' (', 'off ', ' line ', ') '

Having sub word tokenization help programs to identify an affix can give a program additional insight into how unknown words function. From complicated words, like ‘stitching’ to the present tense of the verb ‘stitch’ which helps the machine to identify

the vocabulary easily. It is unlikely that stitching is a word included in many basic vocabularies. Having sub word tokenization, it would be able to separate the word ‘ing’ token to make a valuable word function in the sentence.

Resource consumption and accuracy

Using different tokenization strategies can affect the project in terms of resource consumption and accuracy.

Resources Consumption:

- Character Tokenization results in a higher number of tokens and computational overhead.
- Longer input tokens require more memory during training. Error may occur if the machine has not enough RAM to allocate for the training.
- If the input tokens have exceeded the model’s input limit, the tokens need to be truncated (split), which may lead to increase in computational costs and extended training durations.

Accuracy:

- Word Tokenization results in a higher number of “unknown” tokens due to the OOV problem [1], which will affect the model's understanding of sentences. If a sentence is understood wrongly, the sentiment will also be classified wrongly.
- Character Tokenization eliminates the OOV problem, but overbreaking the input texts makes understanding of language even difficult for the machine [1]. This results in inaccurate predictions.

Modifications for Hotel Reviews Problem

Expected Performance on Classification

In addressing the task of sentiment classification for hotel reviews, particularly when devoid of explicit rating scores, our algorithm confronts unique challenges. This analysis delves into the anticipated performance of the algorithm within this specialised context and proposes strategic modifications to augment its efficacy in discerning sentiment solely from textual content.

Concerning Expected Performance, the algorithm, initially crafted for sentiment analysis with pre-labeled data, may encounter obstacles when applied to the realm of hotel reviews. Its effectiveness hinges on the alignment of language and sentiment expressions in hotel reviews with the patterns observed in the original training data.

A pivotal challenge arises due to the Dependency on Textual Content. In the absence of accompanying ratings, the model must exclusively rely on the text of the reviews to ascertain sentiment. This proves to be a nuanced task, particularly when sentiments are subtly or implicitly conveyed.

Suggested Modifications

To bolster its performance, several Suggested Modifications are proposed. First and foremost, Fine-Tuning with Hotel Review Data is recommended. This involves adapting the model through training it on a dataset specifically composed of hotel reviews labelled for sentiment. This approach enables the model to grasp the intricate language and sentiment nuances inherent to this domain.

Additionally, the suggestion involves Leveraging Pre-trained Models. Commencing with a model pre-trained on extensive text data and subsequently refining it using a smaller dataset of hotel reviews capitalises on the model's existing knowledge base, tailoring it to the intricacies of hotel review sentiment.

Given potential constraints on labelled hotel review data, the proposal extends to Expanding Training Data. Techniques such as rewriting existing reviews or employing language models to generate new examples can compensate for shortages in labelled data, thereby enhancing the algorithm's performance.

Furthermore, to heighten the model's sensitivity to industry-specific sentiments, the recommendation includes Incorporating Industry-Specific Sentiments. This entails integrating a sentiment dictionary tailored to the hospitality industry, facilitating a more precise identification of relevant sentiments within hotel reviews.

Expected Performance with Noisy Ratings

In the realm of sentiment classification, particularly when confronted with inaccurate rating scores within hotel reviews, our classification algorithm encounters significant challenges to both integrity and performance. This analysis aims to address these issues and proposes a comprehensive strategy to enhance model accuracy and resilience in the face of noisy annotations.

Assessing the Anticipated Performance with Noisy Ratings reveals that the impact on **model accuracy** is a foremost concern [13]. Misleading rating scores have the potential to lead the model astray, associating incorrect sentiments with reviews and thereby diminishing its overall accuracy. Additionally, there exists a risk of **overfitting**, wherein the model learns from inaccuracies in the data rather than capturing the true sentiments expressed in the reviews [14].

Strategies for Enhancements

To counteract these challenges, several Strategies for Enhancements are proposed. Firstly, a meticulous Data Review and Correction process is suggested, involving manual inspection and rectification of the data to better understand and mitigate inaccuracies. Moreover, the adoption of Noise-Tolerant Loss Functions, such as focal loss, is recommended to reduce sensitivity to label inaccuracies and prioritise learning from challenging, misclassified examples.

The utilisation of Semi-Supervised Techniques is advocated, starting with a small, accurately labelled dataset for initial training and subsequently expanding the model's learning with a larger set of unlabeled data. Pseudo-labeling, a technique where the model's predictions on unlabeled data are used for training, is highlighted for its potential effectiveness [15].

Furthermore, an Analysis of Prediction Confidence is proposed, focusing on cases where the model's predictions differ from provided labels, especially when the model expresses high confidence. This scrutiny aims to identify potential issues with the labelled data.

Lastly, the recommendation extends to Incorporating Additional Reliable Data from external sources with high-quality sentiment annotations. This infusion of supplementary data is envisioned to strengthen the model's overall performance.

Through the implementation of these adjustments, the model becomes better equipped to navigate the nuances of hotel reviews and effectively address the challenges posed by noisy annotations, thereby fostering more accurate sentiment analysis within this distinctive context.

Contributions by each team member

Question	Contributor(s)
(a)	Wong Yu Hao
(b)	Wong Yu Hao
(c)	Ivan Tan
(d)	Chen Hung-Jui, Wong Yu Hao, Ivan Tan
(e)	Ivan Tan
(f)	Ivan Tan
(g)	Chen Hung-Jui
(h)	Chen Hung-Jui

References

[1] A. BURCHFIEL, "What is NLP (Natural Language Processing) Tokenization?," 16 May 2022. [Online]. Available: https://www.tokenex.com/blog/ab-what-is-nlp-natural-language-processing-tokenization/.

[2] "Summary of the tokenizers," [Online]. Available: https://huggingface.co/docs/transformers/tokenizer_summary.

[3] "WordPiece tokenization," [Online]. Available: https://huggingface.co/learn/nlp-course/chapter6/6.

[4] "Distil*," October 2023. [Online]. Available: https://github.com/huggingface/transformers/tree/main/examples/research_projects/distillation.

[5] B. Lutkevich, "BERT language model," [Online]. Available: https://www.techtarget.com/searchenterpriseai/definition/BERT-language-model.

[6] B. Muller, "BERT 101 🤗 State Of The Art NLP Model Explained," 2 March 2022. [Online]. Available: https://huggingface.co/blog/bert-101.

[7] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,", pp. 3-5 13-15, 11 October 2018.

[8] R. Horev, "BERT Explained: State of the art language model for NLP," 11 November 2018. [Online]. Available: https://towardsdatascience.com/bert-explained-state-of-the-art-language-model-for-nlp-f8b21a9b6270.

[9] "distilbert-base-uncased," Hugging Face, [Online]. Available: https://huggingface.co/distilbert-base-uncased.

[10] "Loss Functions," Machine Learning Cheatsheet. [Online]. Available: https://ml-cheatsheet.readthedocs.io/en/latest/loss_functions.html.

[11] “Bert outputs are different with the same input in training mode · issue #9039 · Huggingface/Transformers,” GitHub.[Online]. Available: https://github.com/huggingface/transformers/issues/9039

[12] A. Thakur, "What's the Optimal Batch Size to Train a Neural Network?," 9 July 2023. [Online]. Available: https://wandb.ai/ayush-thakur/dl-question-bank/reports/What-s-the-Optimal-Batch-Size-to-Train-a-Neural-Network---VmlldzoyMDkyNDU#:~:text=9-,Results%20Of%20Small%20vs.,gave%20us%20the%20best%20result.

[13] E. Beigman and B. B. Klebanov, "Learning with annotation noise," in Proc. Joint Conf. 47th Annu. Meet. ACL and 4th Int. Joint Conf. Nat. Lang. Process. AFNLP, Aug. 2009, pp. 280-287.

[14] J. Li, Y. Wong, Q. Zhao, and M. S. Kankanhalli, "Learning to learn from noisy labeled data," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 5051-5059.

[15] Y. C. A. P. Reddy, P. Viswanath, and B. E. Reddy, "Semi-supervised learning: A brief review," Int. J. Eng. Technol., vol. 7, no. 1.8, p. 81, 2018.