CSCI 4155

Semantic analysis for amazon review

Team 13

Yixiao Yuan B00785417
Kessel Zhang B00809478
Tongqi Liu B00764727

Abstract

With the increasing power of online shopping, people are shopping online more and more. In order to grasp public opinion in time and investigate user feedback, evaluation is indispensable. In order to extract reviews from customer reviews way faster and avoid the phenomenon of scrutiny or partial missing reviews. This project aims to use 4 different models to determine the trend of user reviews, whether it is positive or bad. According to the performance comparison between the models, the analysis concludes which model should be used to meet the existing conditions in different situations.

1. Introduction

It is very important to get the evaluation of others in time, especially as a business. Which foods have received widespread praise from customers, and which products are uninterested. Knowing customer feedback in a timely manner can help merchants formulate relevant policies. Therefore, it is very important to efficiently obtain good or bad reviews from customers. However, most customers do not clearly clarify whether the product they bought is good or bad when they leave a comment. Therefore, not only should we understand the customer’s habits, it is also very important to understand their true intentions from phrases. To this end, our project is to start a series of sentiment analysis on customer reviews so as to help businesses quickly organize the data and lay a solid foundation for future plans.

2. Literature Survey

[1]Amazon fine food review
[2]N-gram
[3]Bag of word model
[4]BERT: Pre-training of Deep Bidirectional
[5]BERT for Natural Language Processing
[6]Understanding BERT — (Bidirectional Encoder Representations from Transformers)
[7] bert-base-uncased
[8]Keras
[9] Gentle Introduction to the Adam Optimization Algorithm for Deep Learning
[10] What is a Confusion Matrix in Machine Learning
[11] scikit-learn.
[12] Classification: ROC Curve and AUC

3. Problem statement

Reviews are the most important source of information for merchants, so we plan to divide the reviews(from amazon dataset[1]) into 5 ratings. The closer to 1 means that this review expresses the customer’s dissatisfaction with the food, it is a bad review, and the closer to 5 means this review expresses the customer's love of the food and is a good comment. Therefore, our goal is to establish different models, not only to predict whether there is a positive or a negative comment based on the text, but also to compare the performance of different models, so as to obtain the usability of each model in different environments.

4. Proposed Technique

In this project, we have adopted two methods to solve our problem, using traditional methods such as tf-idf, bag of word and n-grams to build our model. Another one is BERT, a brand new approach which became popular in recent years. Because the BERT method uses a new technology, the first three models will use the same data processing process while BERT has its own process.

The first part of the data processing flow includes the following three major steps: text normalization, text vectorization and text classification.

4.1 Traditional NLP Method

4.1.1 Text Normalization

Text normalization mainly includes the following parts:

1. Load Data.
2. Tokenization and remove stop words.
3. Remove numbers and other non-letter characters.
4. Lemmatization and POS tagging
5. Result analysis and statistics.

In the first three steps, we have completed the loading of the data and their library and removed any data that is useless for subsequent training. The Lemmatization in the fourth step refers to removing the affixes of the words and extracting the main part of the words. Usually, the extracted words will be words in the dictionary. It is worth noting that different from stemming, the words extracted by stemming may not appear in the words. POS tagging is to tag words, which can be recognized as nouns, verbs, adjectives, adverbs, etc.

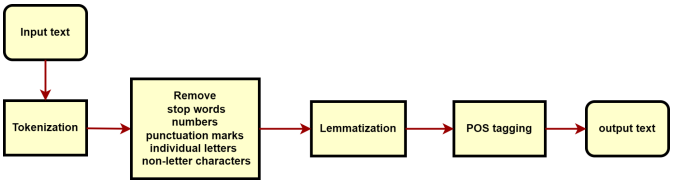

Fig 1. Flowchart of traditional method

4.1.2 Text Vectorization

Text vectorization mainly includes the following parts:

1. n-Gram model[2].
2. Bag-of-words model[3].
3. Term frequency-inverse document frequency(TF-IDF)

The n-gram refers to n words that appear consecutively in the text. The n-gram model is a probabilistic language model based on the (n-1) order Markov chain, which infers the structure of the sentence through the probability of n words appearing

The bag-of-words model is a simplified expression model in natural language processing(NLP). In this model, a piece of text (such as a sentence or a document) can be represented by a bag containing these words. This way of representation does not consider the grammar and the order of the words. The bag-of-words model is widely used in document classification, and the frequency of word occurrence can be used as a feature of training classifiers.

TF-IDF is a statistical method used to evaluate the importance of a word to a document set or one of the documents in a corpus. The importance of a word increases in proportion to the number of times it appears in the document, but at the same time it decreases in inverse proportion to the frequency of its appearance in the corpus.

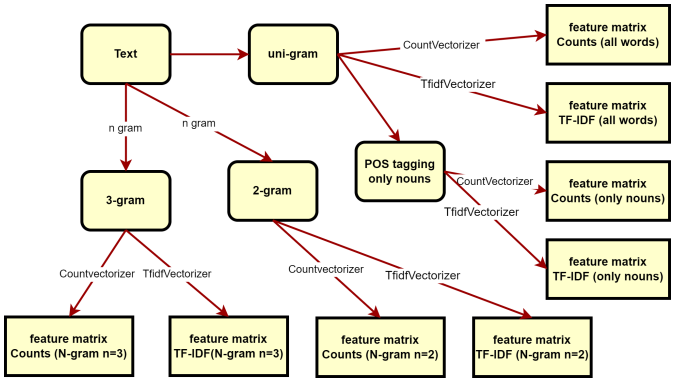

Fig 2. Flowchart details for traditional method

The following figure contains the specific content of the 8 training data sets generated according to the above steps:

	feature matrix	matrix rows	matrix columns	dtype	dtype size(byte)	Memory usage (GB)
0	Counts (all words)	48000	7737	np.uint16	2	0.69
1	TF-IDF (all words)	48000	7737	np.float32	4	1.38
2	Counts (only nouns)	48000	4339	np.uint16	2	0.39
3	TF-IDF (only nouns)	48000	4339	np.float32	4	0.78
4	Counts (N-gram n=2)	48000	28918	np.uint16	2	2.59
5	TF-IDF(N-gram n=2)	48000	28918	np.float32	4	5.17
6	Counts (N-gram n=3)	48000	31541	np.uint16	2	2.82
7	TF-IDF (N-gram n=3)	48000	31541	np.float32	4	5.64

Fig 3. eight training datasets description

4.1.3 Text Classification

Text classification mainly includes the following parts:

1. Model:
    a. Multinomial Naive Bayes Model
    b. Artificial Neural Network Model
    c. Logistic Regression Model
2. Model evaluation (AUC score)

According to a total of 8 training data sets completed in the previous step, call the corresponding library functions to complete the creation of polynomial Bayes, artificial neural networks and logistic regression models.

4.2 BERT method

4.2.1 Extract data from dataset

In this dataset, there are too many positive comments, which may result in an imbalance in the training result. So in this step, a part of positive comments will be dropped to increase the dataset quality.

4.2.2 Labeling data

Similarly, the score 1-5 will be transferred into 1 and 0. And the score of 3 will be dropped because we consider it as a neutral comment.

4.2.3 Clean the data

Firstly, remove all the html tags. Secondly, drop all comments that have length larger than 50. Since we do not want some extremely long comments here.

4.2.4 Build the BERT model

Before processing the text, we should firstly build the BERT and set the proper parameter for the BERT.

BERT is a pre-trained model[4], which means that we do not have to build this model from the very beginning. We will import this model by the tensorflow-hub, and change some hyper-parameter of this model. The BERT model we used in this project is the BERT-base-uncased[7].

The sentence with length larger than 50 has already been dropped from our dataset. So here, we could set the maximum length of a feature to be 55.

After that we will set three inputs for this model. Which are input_ids, input_masks, segment_ids.

4.2.5 Preprocess for BERT

It was mentioned that The BERT model needs three inputs in total[5]. They are separately input_ids, input_masks, segment_ids.

To get the input_ids, we will firstly transfer the words to a number based on their ID provided in the BERT model, then we will add markers ([CLS] and [SEP]) to the start and end of one sentence.

The input_masks use 0s and 1s to illustrate the appearance of the word, and its position in one sentence.

The segement_ids is a list of token type ids to be fed to a model. These are also in the form of 0s and 1s and help to differentiate between the two sequences. The first sequence is the sentiment and the second is the text.

4.2.6 Get the feature of text

The output of our BERT model is the dense vector(feature) of the input text[6]. The output of any sentence will be represented as a 768 dimensions vector. And each word in a sentence has its own feature. Furthermore, the BERT model provides two types of output. They are separately sequential output and pooled output.

The sequential output contains the features of each word in a sentence. For example, for a sentence like "[CLS] I love you [SEP]" , the sequential output has the shape of (5,768). Which means that there are 5 features(each string has one feature), and each feature is in 768 dimensions.

The pooled output is a simplified version of sequential output. The pooled output contains only one feature for each sentence. For example, for a sentence of "[CLS] I love you [SEP]", the dense vector of the string "[CLS]" is considered as the pooled output for the whole sentence.

Commonly, the pooled output is enough for training tasks. So in this project, we will use pooled output for the next step.

4.2.7 Train the model

Now we have the text feature, what needs to be done in the next step is training them.

In this project, we will use keras[8] to train the model. we applied three layers with activation function of "relu", and got the output by "sigmoid".

The optimizor we used is the adam[9] with a learning rate of 0.001.

Theoretically, we will get a vector of two dimensions. Which contains the probability of positive and negative separately.

5. Evaluation Metrics

5.1 Confusion matrix:

A confusion matrix is a summary of prediction results on a classification problem[10].

	Positive	Negative
Positive	TP	FP
Negative	FN	TN

Table 1. confusion matrix

Accuracy, precision, recall and F1 score can be calculated from this confusion matrix.

$$\text{Accuracy} = \frac{(TP+TN)}{(TP+FP+FN+TN)}$$

5.2 Classification report

classification report is a tool for measuring the quality of prediction. We will use the classification report provided by the library of scikit-learn[11].

$$\text{Precision} = \frac{TP}{(TP+FP)}$$

$$\text{Recall} = \frac{TP}{(FN+TP)}$$

$$\text{F1} = 2\times \frac{Precision\times Recall}{Precision+Recall}$$

5.2 AUC-Score:

AUC score is the area under the ROC curve[12]. The result will be smaller and equal than 1 and bigger and equal to zero. The result is good if AUC-Score is close to 1 and vice versa.

6. Performance and Result

6.1 Artificial neural network(Traditional NLP)

In neural networks, batch and epochs are hyperparameters. Among them, the batch defines the number of samples to be processed before updating the internal model parameters; while epochs defines the number of times the entire learning algorithm works in the entire training data set. Therefore, in order to find the best period and batch size, we first initialize the ANN, and finally get the best parameters when epochs = 27 and batch_size = 192.

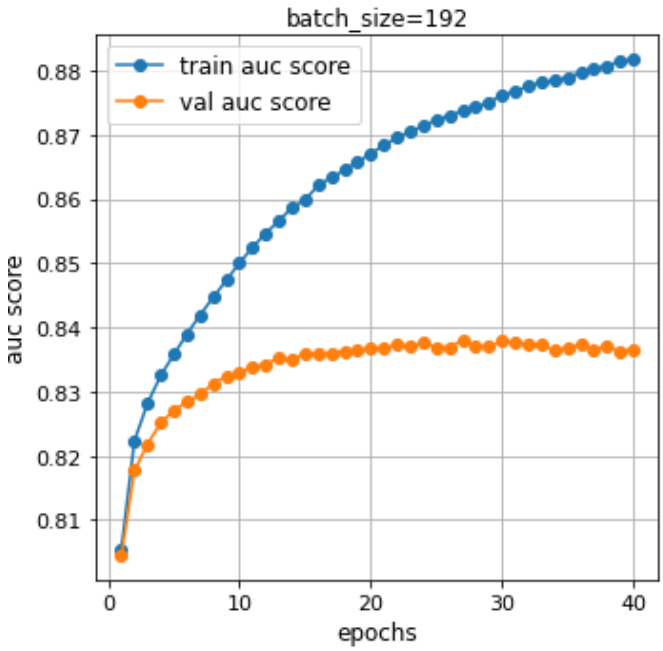

Fig 4. ANN history graph for batch_size=192

After that, after processing the eight data sets generated in the previous step, we got the following results

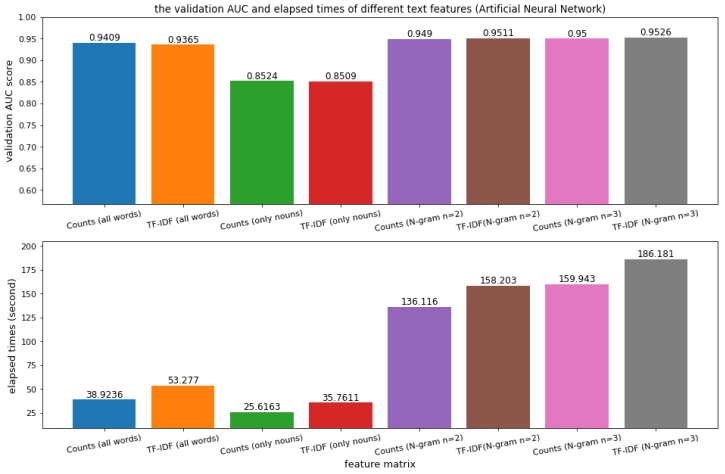

Fig 5. The validation AUC and elapsed time of different text feature(ANN)

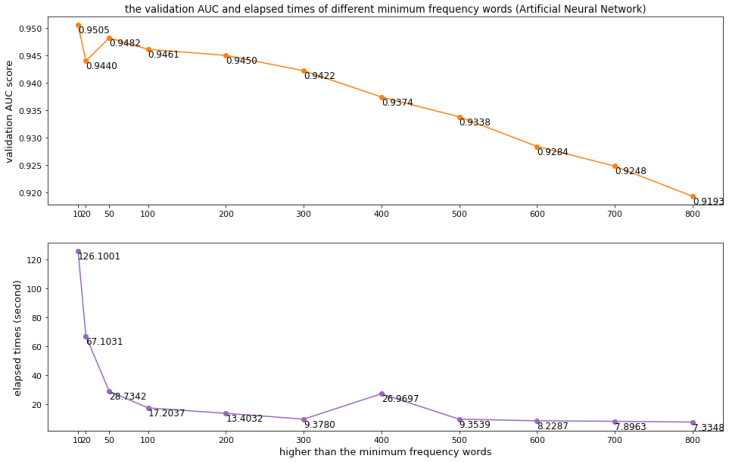

Fig 6. The validation AUC and elapsed times of different minimum frequency words (ANN)

The bar chart shows the accuracy and time-consuming of these eight data sets. The most prominent effect is the use of data sets with n=2 and n=3, but with it, time-consuming increases. The graph shows the time-consuming and accurate changes when we plan to discard some low-frequency words.

Classification report:

	precision	recall	f1-score	support
0	0.87	0.87	0.87	3972
1	0.91	0.91	0.91	5628
accuracy			0.89	9600
macro avg	0.89	0.89	0.89	9600
weighted avg	0.89	0.89	0.89	9600

Fig 7. Classification report for ANN method

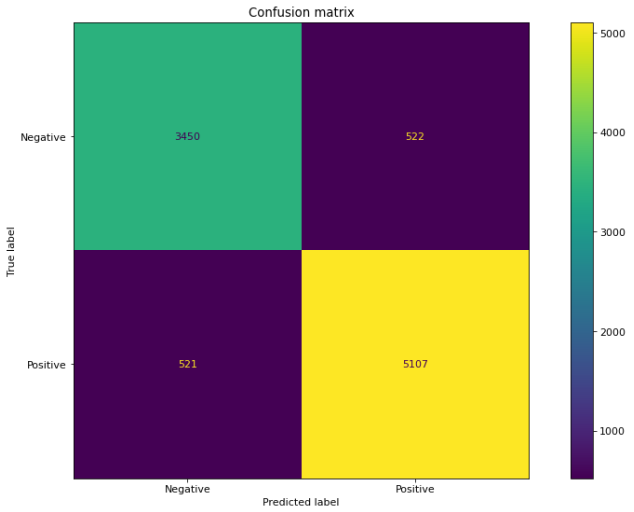

Fig 8. Confusion matrix for ANN method

Base on above graph, we can clearly see that Artificial neural network method have a decent score:

1. Accuracy: 89%
2. Precision: 89%
3. Recall: 89%
4. F1-score: 89%

6.2 Multinomial Naive Bayes(Traditional NLP)

Similarly, the relevant polynomial Naive Bayes is tested on eight data sets. The data set containing n=2 and n=3 still has the highest accuracy, but this time there is a slight gap. We can clearly see that the data set Count(N-gram n=2) consumes the most time while other N-grams consume much less time than this data set.

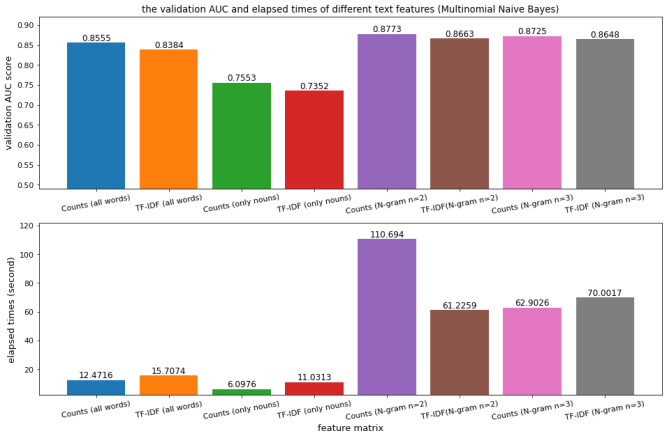

Fig 9. The validation AUC and elapsed times of different text features (Multinomial Naive Bayes)

Similarly, we also considered any changes in accuracy and time-consuming when we discarded some low-frequency words.

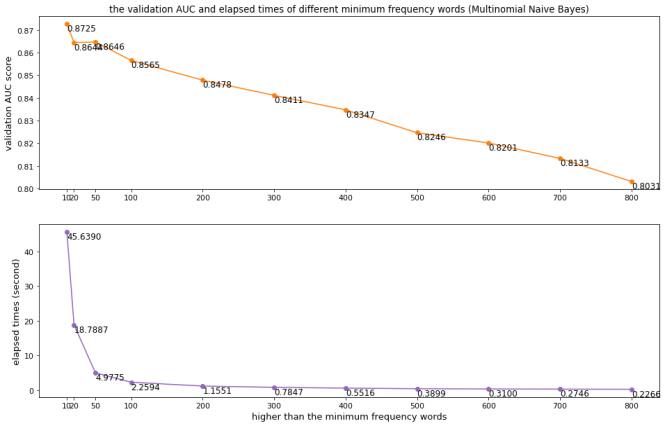

Fig 10. The validation AUC and elapsed times of different minimum frequency words(Multinomial Naive Bayes)

Classification report:

	precision	recall	f1-score	support
0	0.90	0.81	0.85	3972
1	0.88	0.93	0.90	5628
accuracy			0.88	9600
macro avg	0.89	0.87	0.88	9600
weighted avg	0.88	0.88	0.88	9600

Fig 11. Classification report for Multinomial Naive Bayes

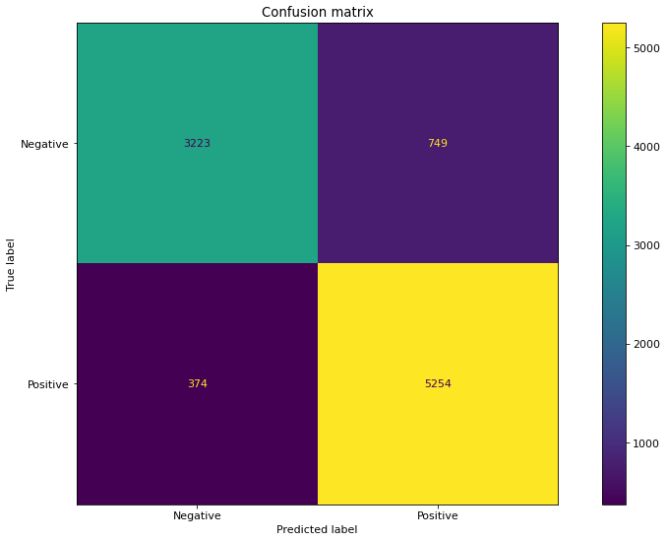

Fig 12. Confusion matrix for Multinomial Naive Bayes

Base on above graph, we can clearly see that Multinomial Naive Bayes Classifier method have a good score:

1. Accuracy: 88%
2. Precision: 89%
3. Recall: 87%
4. F1-score: 88%

6.3 Logistic Regression(Traditional NLP)

Finally, when we used logistic regression to test these eight data sets, we got new findings. Although in terms of accuracy, the data set with n-gram is still dominant, at the time-consuming level, the n-gram data set with TF-IDF takes much less time than the data set with the same value of n.

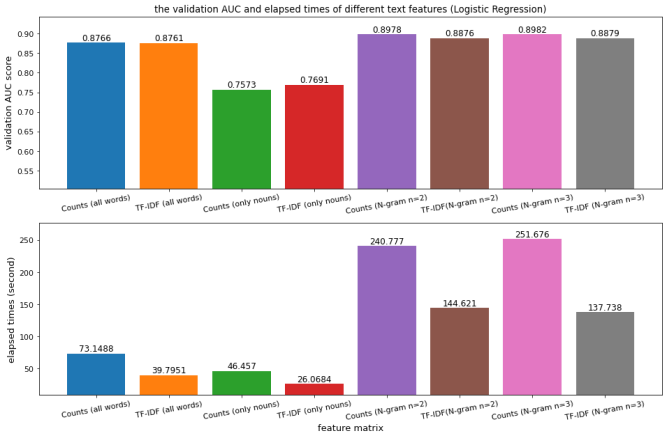

Fig 13.The validation AUC and elapsed times of different text features(Logistic Regression)

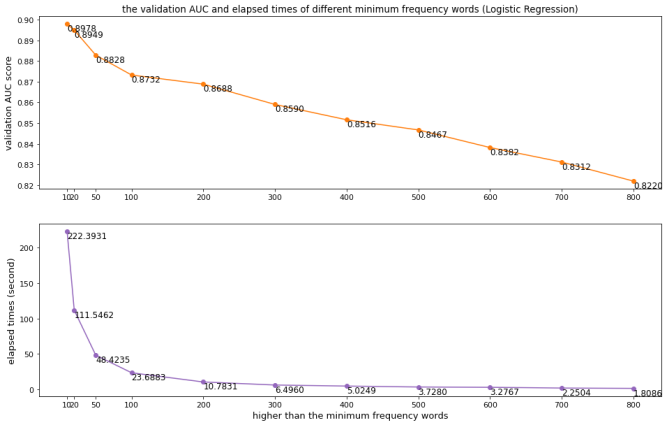

Fig 14.The validation AUC and elapsed times of different minimum frequency words(Logistic Regression)

```
Classification report:
              precision    recall  f1-score   support

           0       0.88      0.88      0.88      3972
           1       0.91      0.92      0.92      5628

    accuracy                           0.90      9600
   macro avg       0.90      0.90      0.90      9600
weighted avg       0.90      0.90      0.90      9600
```

Fig 15. Classification report for Logistic Regression method

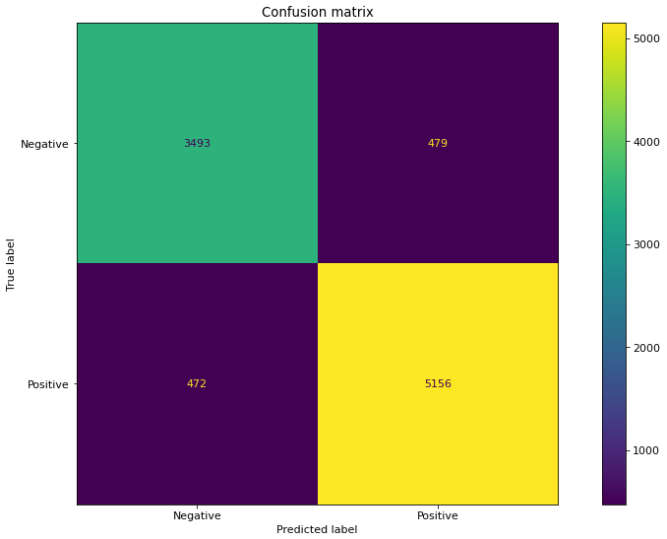

Fig 16. Confusion matrix for Logistic Regression method

Base on above graph, we can clearly see that Logistic Regression method occupy a little advantage:

1. Accuracy: 90%
2. Precision: 90%
3. Recall: 90%
4. F1-score: 90%

6.4 BERT method

Firstly, let us see the performance graph of this method. The x-axis is the number of epochs in the training, the y-axis is the AUC score we get for each epoch.

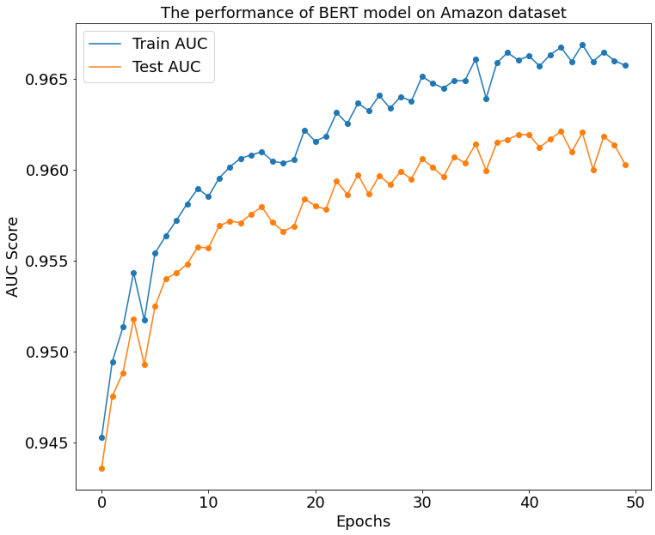

Fig 17. The AUC performance of BERT model

Also, to see the result more clearly, we will introduce the classification report to illustrate the performance.

Classification report:

	precision	recall	f1-score	support
0	0.89	0.82	0.86	1489
1	0.90	0.94	0.92	2557
accuracy			0.90	4046
macro avg	0.90	0.88	0.89	4046
weighted avg	0.90	0.90	0.90	4046

Fig 18. The classification report of BERT model

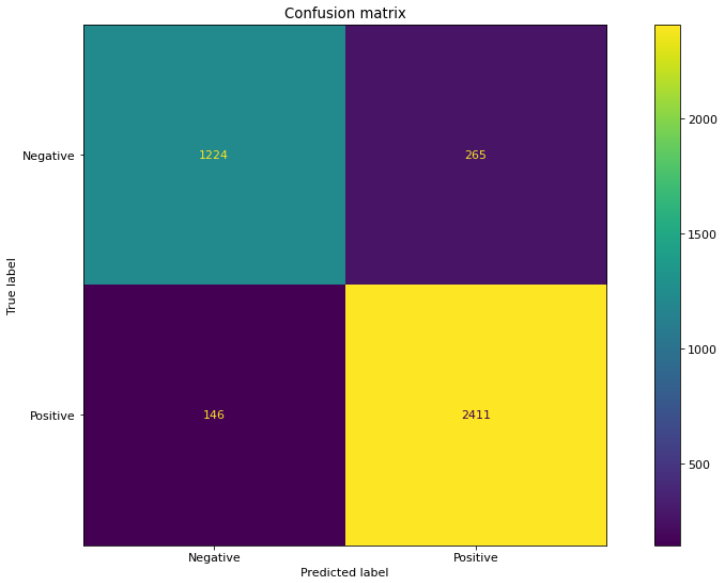

Fig 19. The confusion matrix of BERT model

From the picture above, we could observe that the BERT method have good score in difference matrices:

1. AUC: 96%
2. Accuracy: 89%
3. Precision: 89%
4. recall: 88%
5. f1-score: 89%

7. Conclusion and future Scope

We have implemented both the traditional NLP method and advanced NLP method(BERT). Among these methods, different algorithms perform differently.

By comparison, the BERT method has achieved the highest AUC score(approximately 96%). Which means that this model will perform better than the traditional method.

However, when we come to the comparison of accuracy, the advantage of BERT is not that obvious. Both of these two methods have similar accuracy scores, about 90%. I think this is because of the imbalance of the test set. Since we have the different numbers of the “Positive” and “Negative” reviews in the train set and test set. Which results in some error in the accuracy matrices.

Meanwhile, the running time of different models should also be taken into account. Although the BERT model gives a good result, it is too slow. By contrast, both the TF-IDF(all words) and BOW(all words) have less cost of time(only about 6-10 seconds in the lowest set).

Generally speaking, if we are seeking for the highest AUC performance, the BERT model is the best one. But if the running time should be considered, the TF-IDF and BOW is also a good choice.

For the future scope, there are still many things that could be done to further improve the quality of our project.

1. Tune the parameters for our methods to try to get a better result.
2. Apply more layers on our model(DNN), to see if more layers give higher accuracy. Commonly, a deeper model has better performance.
3. Apply more advanced BERT models. There have been multiple versions of the BERT model that are suitable for different tasks. Try them.
4. Preprocess the dataset in another way. For example, there might be different results if we set a score of 3 to “Positive” instead of “Neutral”.
5. Apply more regularization methods on our model to prevent overfitting/get a higher score.

8. Reference

[1]*Amazon Fine Food Reviews*. (n.d.). Kaggle.com. https://www.kaggle.com/snap/amazon-fine-food-reviews

[2]Wikipedia Contributors. (2019, February 27). *n-gram*. Wikipedia; Wikimedia Foundation. https://en.wikipedia.org/wiki/N-gram

[3]*Bag-of-words model*. (2020, September 21). Wikipedia. https://en.wikipedia.org/wiki/Bag-of-words_model

[4]Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. ArXiv.org. https://arxiv.org/abs/1810.04805

[5]*BERT for Natural Language Processing |All You Need to know about BERT*. (2021, May 27). Analytics Vidhya. https://www.analyticsvidhya.com/blog/2021/05/all-you-need-to-know-about-bert/

[6]Vajpayee, S. (2020, August 6). *Understanding BERT — (Bidirectional Encoder Representations from Transformers)*. Medium. https://towardsdatascience.com/understanding-bert-bidirectional-encoder-representations-from-transformers-45ee6cd51eef

[7] *bert-base-uncased · Hugging Face*. (n.d.). Huggingface.co. https://huggingface.co/bert-base-uncased

[8]Keras. (2019). *Home - Keras Documentation*. Keras.io. https://keras.io/

[9]https://www.facebook.com/jason.brownlee.39. (2017, July 2). *Gentle Introduction to the Adam Optimization Algorithm for Deep Learning*. Machine Learning Mastery. https://machinelearningmastery.com/adam-optimization-algorithm-for-deep-learning/

[10]https://www.facebook.com/jason.brownlee.39. (2018, May 30). *What is a Confusion Matrix in Machine Learning*. Machine Learning Mastery.

[11] scikit-learn. (2019). *scikit-learn: machine learning in Python — scikit-learn 0.20.3 documentation*. Scikit-Learn.org. https://scikit-learn.org/stable/

[12]Google. (2019). *Classification: ROC Curve and AUC | Machine Learning Crash Course*. Google Developers. https://developers.google.com/machine-learning/crash-course/classification/roc-and-a

Appendix A: Team Contribution

Component	Team member
Data processing	Kessel, Yixiao
Method of BERT	Kessel
Method of tf-idf, bow	Yixiao
Naive Bayes Logistic Regression	Yixiao, Tongqi
Report	Yixiao, Tongqi, Kessel

Appendix B: Contribution Percentage

members	percentage
Yixiao, Yuan	**34%**
Kessel, Zhang	**33%**
Tongqi, Liu	**33%**