DM2583 Big Data in Media Technology
Project Report
Sentiment Analysis of Big Data on Social Media By Using Supervised Approaches

GROUP 6
Members: Xin Sun, Yuan Wo, Yuxin Meng

Abstract

Sentiment analysis is a common application of natural language processing (NLP). It is the Analysis, processing, induction, and reasoning of subjective text with emotion, the method of quantifying qualitative data by using some emotion score index. In natural language processing (NLP), sentiment analysis is a typical text classification problem, that is, the text that needs sentiment analysis is divided into its own category. Existing research has produced several techniques that can be used to multitask emotion analysis, such as using some supervised machine learning methods (such as Support Vector Machine, Naive Bayes, etc.) and feature extraction. In the past few decades, with the widespread use of social media, people have expressed their opinions and emotions about various topics or things on various websites or applications. In this project, we'll discuss several sentiment analysis algorithms and compare their performance using Amazon's past movie reviews as data. We used Naive Bayes, Support vector machine, random forests, and long short-term memory (LSTM) to implement the sentiment analysis algorithm. Data preprocessing is an essential part, and effective data processing can improve the performance of the sentiment analysis model. So this article will also show you how we process and analyze data. By using the relevant libraries in Python, we can train the model. The results show that the prediction accuracy of Linear Support Vector Machine is higher and the overall performance is better.

1 Introduction

Social media has become a very important part of people's daily lives. People use social media as a tool to share their experiences, opinions and communicate with others. More than that, social media has also made it possible for people to comment on the products they have experienced, and this includes the film industry. Most people generate their own reviews of a film after watching it, and social media provides a platform for people to do so. As one of the world's largest online suppliers, Amazon naturally has a large number of users who leave their own feelings and ratings on products, and the same is true of the movie platform that Amazon offers. On it, it is possible to see what kind of feedback the audience will give about the film, what kind of review score they will give. Such data contain their emotions and is perfect for sentiment analysis.

Sentiment analysis (SA) is a computational study of people's opinions, attitudes and emotions about something. This something can be an event, a specific something or a topic, etc. These are most likely to be covered by comments. Sentiment analysis identifies the emotions expressed in a text and then analyses them[15]. Sentiment analysis is essentially a classification problem, a type of supervised learning, and can use classical machine learning models such as SVM, random forest, logistic regression, etc. We can also use deep learning neural networks for emotion sharing.

2 Related Work

At present, the method of sentiment analysis can be divided into 3 methods which are sentiment analysis method based on sentiment dictionary, sentiment analysis method based on traditional machine learning and sentiment analysis method based on deep learning respectively.

2.1 Method based on sentiment dictionary

The method based on sentiment dictionary refers to the division of sentiment polarity at different granularities according to the sentiment polarity of sentiment words provided by different

sentiment dictionaries. Most of the existing sentiment dictionaries are artificially constructed. According to the different granularity of the division, the existing sentiment analysis tasks can be divided into words, phrases, attributes, sentences, texts and other levels. For the construction of sentiment dictionary, the earliest is Senti Word Net[2] sentiment dictionary, which combines words with the same meaning according to WordNet, and assigns positive or negative polarity scores to the words. The sentiment polarity score can truthfully reflect the users' emotional attitudes.

2.2 Method based on machine learning

Using machine learning methods for text sentiment analysis is a popular research direction in recent years, where test data are identified by training data and then feature extraction is performed. A text sentiment analysis model is generated by model training, and then text sentiment analysis is performed.

Pang et al[18] in 2002 used plain Bayesian, maximum entropy and support vector machine in text sentiment analysis for comparison and found that text sentiment analysis using the SVM achieves optimal results. Hajmohammadi[8] used standard machine learning techniques the SVM and Naive Bayesian to automatically classify movie reviews in the Persian language as positive and negative and found that the SVM classifier achieved higher accuracy than plain Bayes in movie reviews in the Persian language. As technology has evolved, the use of deep learning neural networks for sentiment analysis has been implemented.

2.3 Method based on deep learning

With the development of the sentiment analysis method based on deep learning, the basic sentiment analysis method of a single neural network was gradually improved, and then introduced sentiment analysis of the mixed (combination, fusion) neural network, sentiment analysis of the attention mechanism and sentiment analysis using pre-trained models.

2.3.1 Single neural network

In 2003, Bengio et al.[4] proposed a neural network language model, which uses a three-layer feedforward neural network to model. The neural network is mainly composed of an input layer, hidden layer, and output layer. Each neuron in the input layer of the network represents a trait. The number of hidden layer and hidden layer neurons is manually set, and the output layer represents the number of classification labels. This method can effectively solve the problem in traditional sentiment analysis methods which is about ignoring the problem of contextual semantics.

Typical neural network learning methods include convolutional neural network, recurrent neural network[20], long and short-term memory network and so on. Many of these scholars have found that the LSTM model is effective in processing long-term data and learning long-term dependence in sentiment analysis. It is one of the most suitable neural networks for deep learning sentiment analysis. Hassan et al [9] used several deep learning methods for sentiment analysis of short texts in 2017, which include the LSTM neural network method.

2.3.2 Sentiment analysis of mixed (combination, fusion) neural network

This method is to combine and improve these methods after considering the advantages of different single neural network methods and using them for sentiment analysis.

Some examples are given below. Taking full account of the advantages of recurrent neural networks and convolutional structures, Madasu et al.[13] proposed a sequential convolutional attention recurrent network (SCARN), compared with the traditional CNN and LSTM methods, SCARN has better performance; Luo Fan et al.[**?**] used joint recurrent neural networks and

convolutional neural networks to propose a multi-layer network model HRNN-CNN, The model uses two-layer RNN to model the text and introduces it into the sentence layer to realize the sentiment classification of long texts and so on.

2.3.3 Sentiment analysis of the attention mechanism

The earliest application of the attention mechanism was in the field of visual images. In the beginning, researchers used the attention mechanism on the RNN model to achieve image classification[16]. Later, Bahdanau et al. [3] applied the attention mechanism to machine translation tasks, which also meant that the attention mechanism began to be applied to the field of natural language processing. The attention mechanism can expand the capabilities of neural networks, allowing more complex functions to be approximated, that is, focusing on specific parts of the input. By using this mechanism in a neural network, the performance of natural language processing tasks can be effectively improved. For example, Yang et al.[21] first proposed a collaborative attention mechanism that alternately modelled target-level attention and context-level attention, and realized aspect sentiment analysis by transferring the target to the contextual representation of keywords. The experiments on the SemEval2014 dataset and Twitter dataset show that this method is better than the traditional neural network method with an attention mechanism.

By adding the attention mechanism to the deep learning method for the research of sentiment analysis tasks, it can better capture context-related information, extract semantic information, prevent the loss of important information, and effectively improve the accuracy of text sentiment classification.

2.3.4 Sentiment analysis using pre-trained models

The pre-trained model refers to the model that has been trained with the data set. By fine-tuning the pre-training model, better emotion classification results can be achieved. The latest pre-training models are ELMo, BERT, XL-NET, ALBERT, etc.

Peters et al.[19] proposed a new method ELMo at the NAACL conference (The North American Chapter of the Association for Computational Linguistics) in 2018. The method uses a two-way LSTM language model, consisting of a forward and a backward language. The objective function in ELMo is to take the maximum likelihood value of the language model in these two directions. Compared with the traditional word vector method, the advantage of this method is that each word corresponds to only a one-word vector.

In October 2018, Google proposed a new method based on BERT[6], which uses a two-way transformer mechanism for the language model, fully taking into account the context and semantic information of words. BERT uses WordPiece embedding as a word vector in the input of the model and adds a position vector and a sentence segmentation vector. Many scholars have also made some fine-tuning improvements based on BERT and have had good results. we won't go into too much detail here.

3 Research question

The project topic is regarding to sentiment classification of social media. Social media has many directions, such as food reviews, movie reviews, book reviews, and open forums. Since the three members of the group usually like to watch movies, they unanimously decided that our choice of social media should be positioned in movie reviews.

So, based on the general direction, we will use a dataset of movie reviews on Amazon from 1997 to 2012[14]. This dataset has close to 8 million movie reviews, which is large enough for us to train and test the models.

Also according to existing studies, we became interested in the performance of these different methods used for sentiment analysis. So, our project chose some of the algorithms mentioned above which are from method based on machine learning and method based on deep learning to perform sentiment analysis on a dataset of our choosing. Finally, we can conclude that our research question is:

For movie review dataset, which model performs best based on accuracy and time?

The main goal of the project is to compare the performance of these models trained with different algorithms and to find which algorithm gives the best results for sentiment analysis. When using classical machine learning models, we usually develop a series of sentiment lexicons and rules to disassemble the text, extract keywords, and then correspond the keywords to the sentiment and determine the sentiment of the text based on this. The algorithm we select based on the method based on deep learning only focuses on the single neural network which can be the most representative and lowest cost deep learning method.The single neural network we chose is Long Short-term Memory (LSTM).

4 Hypothesis and Project Plan

According to the background introduced in related work, it is not difficult to find that traditional machine learning methods have many excellent models in sentiment analysis, but in general, deep learning methods perform better. Especially for the direction of text sentiment analysis, the Long and Short-term Memory model can be considered as the most suitable single neural network. Therefore, for the several models we selected, the final assumption is:

We assume Long Short-Term Memory (LSTM) will perform best.

Then we conclude that our project plan can be listed as the following points.

1. Data prepossessing – Finally get labeled data
2. Model construction – Train different algorithms with processed data to get the best models respectively.
3. Evaluation – Test the models by using at least one million test data, and then measure the performances of classifiers by using various methods.
4. Results and analysis – Organize the evaluation results and analyze the results of different models.
5. Conclusion and discussion – Compare the results and hypotheses, analyze the reasons for consistency and inconsistency, discuss the problems encountered and the directions for improvement.

5 Datasets

5.1 Data Collection

When training a model, there is a need for data that can effectively distinguish sentiment.For our project, we have chosen to review the film. First of all, there are a large number of such reviews on social media platforms and the ratings that reviewers give to films can represent the general sentiment of the reviews they leave. Therefore, we chose a dataset of about 8 million movie reviews, from which we randomly selected a portion as the training dataset, and then selected at least 1 million data from the rest as the test dataset. Based on the ratings, we can divide the texts into three categories: positive, negative and neutral. And following is the structure of the data units in the dataset.

```
product/productId: B00006HAXW
review/userId: A1RSDE90N6RSZF
review/profileName: Joseph M. Kotow
review/helpfulness: 9/9
review/score: 5.0
review/time: 1042502400
review/summary: Pittsburgh - Home of the OLDIES
review/text: I have all of the doo wop DVD's and this one is as good or better than the
1st ones. Remember once these performers are gone, we'll never get to see them again.
Rhino did an excellent job and if you like or love doo wop and Rock n Roll you'll LOVE
this DVD !!
```

Figure 1: Data form

5.2 Data Prepossessing

As the original structure of the dataset is not suitable for direct use in model training, an important step is the pre-processing of the data, which generally involves transforming the data into a more processing-friendly format, reducing the interference of unnecessary data, and analysing the text for different emotions.

Processing data from the original Amazon movie reviews is hard work because of the format of its text. It will take a lot of time if we process the whole dataset. Because of this, we choose another dataset with a better format for us to process. Then, we use several preprocessing methods to clean the data we get.

5.2.1 Remove Noisy

Reviews from people always contain special characters, URL's/HTML tags, emotions and mentions which have no help for sentiment analysis. Thus, the first thing of processing data is to remove these 'noisy data' and also map the contractions into formal writing. To do this, we use a custom function called 'comment_to_words'.

5.2.2 Remove Stop-words

Stop-words are words that are used commonly like 'and', 'are' and other similar words. These words do not have much relevance to the expression of emotion, so we can remove them from the comments. And then the number of features in our classifier can be reduced. As for how to determine which words are stop-words, we chose to use the stop-words corpus of the NLTK, which contains 127 words.

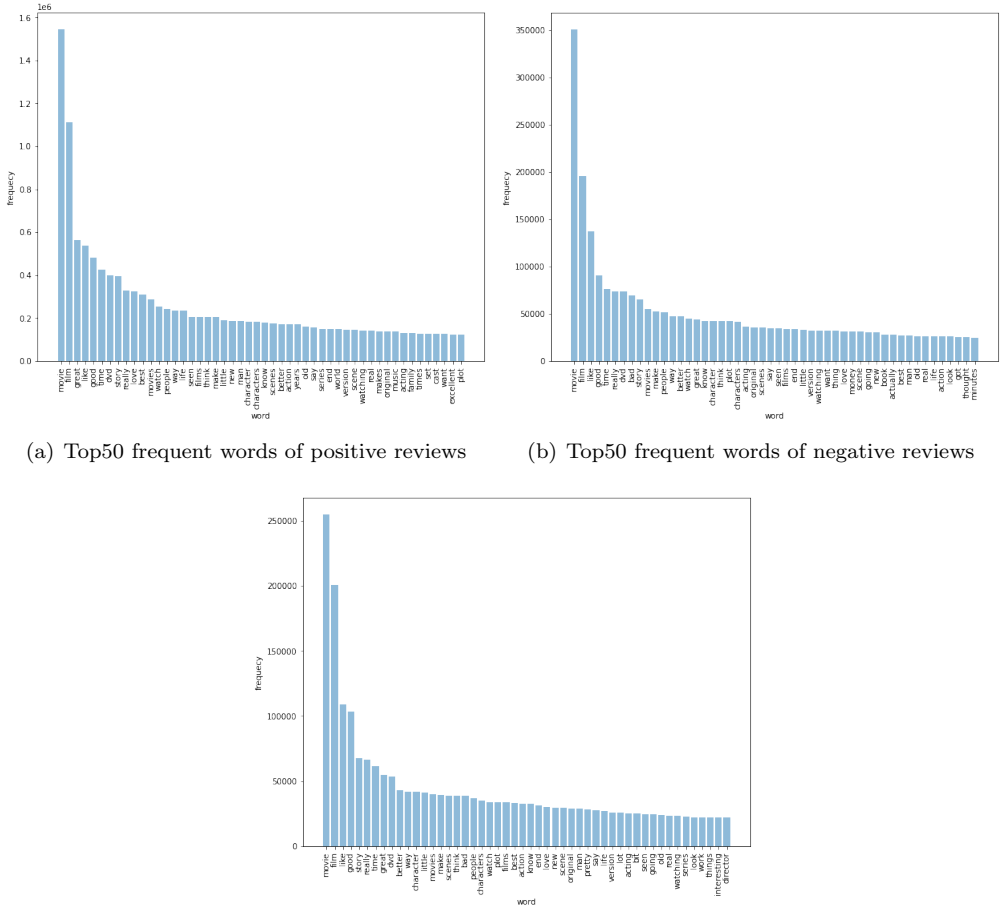
5.2.3 Remove NAN Value

We find there are some raws of the 'score' column have values that is not a number when we try to change the scores into labels, these NAN value data cannot be used for sentiment analysis categorisation, so we need to remove them. We chose to use the function of the Pandas library to remove those NAN raws from our dataset.

5.3 Data Analysis

After prepossessing the data, we also performed a data analysis on the processed data set. Such analysis can help us better understand the data set and the classification of the data. The analysis of the data is mainly reflected in the frequency analysis of the words mentioned in the comments of different emotions. Such analysis can help us visually see the relationship between the emotion of the word and the emotion of the comment.

As shown in Figure 2, except for the words describing the movie, words that directly express emotions, for example, words such as "great", "good", "best", and "love" are mentioned



(a) Top50 frequent words of positive reviews

(b) Top50 frequent words of negative reviews

(c) Top50 frequent words of neutral reviews

Figure 2: Top50 frequent words in 3 categories

frequently in positive comments. On the other hand, the words "bad" and "little" appear more frequently in reviews that give a low score to the movie. In addition, we can also find that in the more neutral comments, the two words "good" and "bad" appear more frequently, which is in line with expectations. However, we can also see that in relatively negative comments, "like" and "good" appear to be words that express positive emotions. This may be a little strange. If we analyze it in combination with some expression habits, it is understandable. For example, some people prefer to use the negative word "not" with such a word to express their negative attitudes.

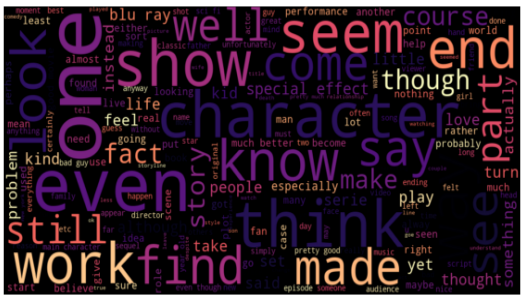
We also present these analysis results in the form of a wordcloud. This also makes the results of the above analysis more straightforward. As can be seen in the word cloud, positive words are clearly seen in the positive comments. The adjectives in neutral comments are not so obvious.

5.4 Feature Extraction

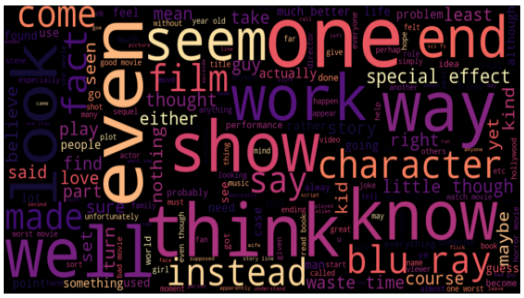
5.4.1 Count Vectorizer

In order to extract features, we choose to use the text feature extraction function CountVectorizer(). The function takes into account the frequency of each word, and then forms a feature matrix, with each line representing word frequency statistics for a training text. By converting text words into vectors, we can get the frequency of the keywords about emotion, which

(a) Words Cloud of positive reviews



(b) Words Cloud words of negative reviews



(c) Words Cloud words of neutral reviews

Figure 3: Words Cloud of 3 categories

can be used to train the model. When no prior dictionary exists, Countvectorizer acts as an Estimator to extract words for training and generates a CountVectorizerModel for storing the corresponding vocabulary vector space. During the training of the CountVectorizerModel, the CountVectorizer will select words in the corpus-based on their frequency ranking from highest to lowest and convert the words in the text into a word frequency matrix.

5.4.2 Tokenization

In sentiment analysis, the text used is 'unstructured data', which we need to transform into 'structured data', and word separation is the first step we need to take. Tokenization is used to split a sentence into individual words, keywords or phrases called tokens. The methods will tokenize all reviews into sequences of words and then give each distinct word a unique number. This process is similar to constructing a dictionary of words. The final representation of the reviews would be an array of numbers showing the index of the word in the dictionary.

6 Methodology

For sentiment classification, there are two basic methodologies: Symbolic techniques and Machine Learning techniques.[5] The way we used for this project is Machine learning technology. In this section, we briefly introduce the general machine learning technologies for sentiment classification. Next, we will explain the specific algorithms we used and claim why we chose them. Here we use 3 algorithms which are Naive Bayesian, Support Vector Machine and Long short-term Memory.

6.1 Sentiment classification by using machine learning approach

Machine learning method generally divided data into training data and testing data. The training data includes the features and the labels, then the machine will then generate the classification model about such data which to classify the label based on the features. And the test data is used to test the performance of the classification model which is about predicting the labels with unknown features. For sentiment classification, machine learning approach uses unsupervised, weakly supervised or fully supervised learning to construct a model from a large training corpus.[5] The common supervised learning is Naive Bayesian and Support Vector Machine. Also, the last approach LSTM is also a supervised learning neural network.

6.2 Naive Bayesian

Naive Bayesian is based on the Bayesian Theorem and also assumes that all the features you use are independent. It means Naive Bayesian doesn't consider the relationship between features.[17].In the case of a given category y, we can get the conditional probability of Naive Bayes is as the following equation:

$$P = (X|Y = y) = \prod_{i=1}^{d} P(x_i|Y = y) \tag{1}$$

Here X is the feature vector which is defined as $X = \{x_1, x_2, ...x_m\}$. And in the sentiment classification, the features can be the emotions or keywords.

In our Naive Bayesian model, we set the labels as -1,0,1 for negative, neutral and positive attitudes respectively.

6.3 Support Vector Machine

Support vector machines (SVMs) have been shown to be highly effective at traditional text categorization, generally outperforming Naive Bayes.[11].In sentiment analysis, SVM uses a hyperplane to separate the labels, so that we can classify the different sentiment categories. The discriminate function is shown as follows:

$$g(X) = w^T \phi(X) + b \tag{2}$$

Here, 'X' means the feature vector, 'w' means the weight vector and 'b' means the bias vector. 'w' and 'b' are all learned from the training process. SVMs is to map the data from high dimensional space to low dimensional space, so here $\phi()$ means such mapping. Also, SVMs can also use linear or no linear kernel to do the classification. Here we also will use such two kernels in our model comparisons.

As the same as the setting in Naive Bayesian, we set the labels as -1,0,1 for negative, neutral and positive attitudes respectively.

6.4 Long Short-term Memory

In the sentiment analysis area, neural network is also a very common method. For the text sentiment, we need a good language model which shall consider the sequential relationship between words. Mikolov then proposed a neural network which is called Recurrent Neural Network (RNN)[10]. Such a network consists of three parts which are input layer, hidden layer and output layer. At the time t', the input layer and the hidden layer have been aggregated as a new input layer to calculate the hidden layer at the time t'. So, the hidden layer can store the information about the previous word which satisfies the requirement of the sentiment analysis. In theory, RNN can cover the chronological structure of the entire text to deal with long-term dependence issues, but in practice, it faces some problems. If the interval between the relevant

information of the text and the current position to be predicted becomes larger, the expansion layers will be too many in the back propagation of the time optimization algorithm (BPTT), and the historical information will be lost and gradient attenuation during training. To solve such problems, here Long Short-term Memory has been introduced.

Different from RNN, LSTM network uses LSTM units to replace nodes in the hidden layer. The structure of LSTM unit is to use three gates to control the use and update of text history information, which are input gate, forget gate and output gate. The structure diagram is shown in Figure 4[7].

The working process of LSTM is like this. First, the forget gate determines which infor-

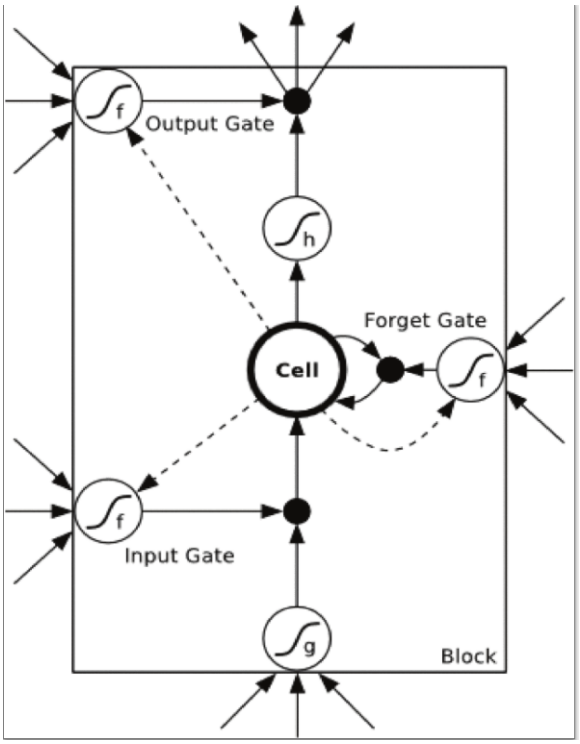


Figure 4: A LSTM Unit Structure

mation from the cell state should be discarded. Then enter the gate to determine the new information stored in the cell state. The third step is to update the old cell state, and use the input gate and forget gate information to calculate the updated value of the cell state. Finally, the output gate determines the value of the output.[1]

The calculation process can also be defined as 4 parts based on the working process of LSTM.[12]

1. Compute the values of forget gate and input gate.
2. Update the state of the LSTM Unit.
3. Compute the value of output gate.
4. Update the state of the entire cell.

In order to compute the values of Input Gates, we use the following equation.

$$a_l^t = \sum_{i=1}^{I} w_{il} x_i^t + \sum_{h=1}^{H} w_{hl} b_h^{t-1} + \sum_{c=1}^{c} w_{cl} s_c^{t-1}$$
$$b_l^t = f(a_l^r)$$

For the calculation of the Forget Gates:

$$a_\emptyset^t = \sum_{i=1}^{I} w_{i\emptyset} x_i^t + \sum_{h=1}^{H} w_{h\emptyset} b_h^{t-1} + \sum_{c=1}^{c} w_{c\emptyset} s_c^{t-1}$$
$$b_\emptyset^t = f(a_\emptyset^t)$$

The update of the cell:

$$a_c^t = \sum_{i=1}^{I} w_{ic} x_i^t + \sum_{h=1}^{H} w_{hc} b_h^{t-1}$$
$$s_c^t = b_\emptyset^t s_c^{t-1} + b_l^t g(a_c^t)$$

The calculation of the Output Gates:

$$a_t^l = \sum_{i=1}^{I} w_{iw} x_i^t + \sum_{h=1}^{H} w_{hw} b_h^{t-1} + \sum_{c=1}^{c} w_{cw^{S_C^t}}$$
$$b_w^t = f(a_w^r)$$

The final update of the whole cell output:

$$b_c^t = b_w^t h(s_c^t)$$

For the label setting in our LSTM model, we make a small change for convenience which set the labels as 2,0,1 for negative, neutral and positive attitudes respectively.

7 Evaluation

Accuracy is often used as a criterion to assess the performance of a model, however, due to the test dataset used, it may not be balanced across all types of data, so in order to reduce this effect, there are several other aspects of the model performance that need to be assessed. So in addition to this, we also use the confusion matrix, the Area Under Curve(AUC), and the time taken to train the model to measure the performance of the model in various ways.

7.1 Classification Accuracy

Classification accuracy represents the percentage of text types that are correctly determined by the model, which means that the higher the accuracy, the greater the number of correct classifications of the data. Although the accuracy calculation is simple and the time complexity is low, however, in the case of binary classification and unbalanced positive and negative examples, the accuracy evaluation is much less informative.

7.2 Confusion Matrix

A confusion matrix, also known as an error matrix, is a scenario analysis table that summarises the prediction results of a classification model in machine learning. The records in a

data set are summarized in matrix form according to two criteria: the true category and the category predicted by the classification model and are represented as a matrix with n rows and columns. Each column of the confusion matrix represents the predicted category and the total of each column represents the number of data predicted to be in that category. Each row represents the true attribution category of the data and the total of the data in each row represents the number of data instances in that category.

So once we have the confusion matrix of the model, we can judge the performance by looking at the values of the areas on the confusion matrix where the predicted values correspond to the same values as the true values. The higher the value, the higher the number of correct predictions.

7.3 The Area Under Curve

AUC (Area Under Curve) is defined as the area under the ROC curve. The ROC curve, known as the receiver operating characteristic curve, is a curve based on a series of different dichotomies (cut-off values or decision thresholds), with the true positive rate (sensitivity) as the vertical coordinate and the false positive rate (1-specificity) as the horizontal coordinate.

AUC is an evaluation metric that measures the merit of a binary classification model and indicates the probability of a positive predicted case coming before a negative case. This method is simple, intuitive, graphically observable to analyze the accuracy of the learner, and allows judgments about model performance to be made directly from observation. As the ROC curve is generally above the straight line y=x, the AUC can range between 0.5 and 1. The closer the AUC is to 1.0, the higher the truthfulness of the test method; when it is equal to 0.5, it is the least truthful and has no application value.

Since AUC can only be used in binary classification problems, here we have three labels, so we define if the labels are classified as the positive label, we consider the result is true, others will be false. In this way, we can transform such a multi-label problem into a binary classification problem.

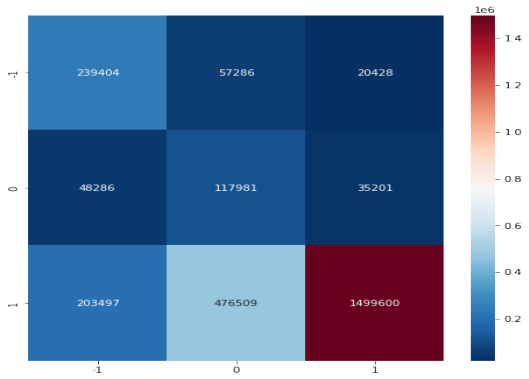
7.4 Training Time

In addition to the above frequently used metrics to measure model performance, we also consider the time taken to train the model as one of the metrics. This indicates to some extent the time complexity of the model algorithm, and if the accuracy of the models is similar to each other, we can assume that the model that takes less time to train is better.

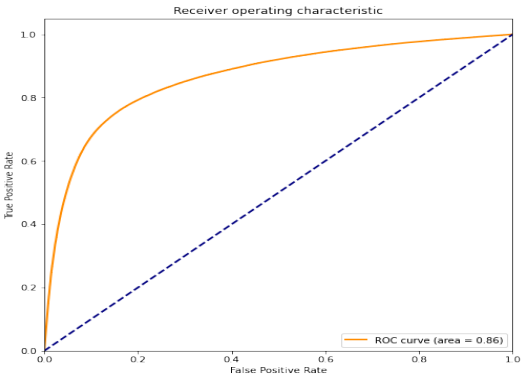
8 Result

8.1 Naive Bayes Classifier

We use the Area Under the Receiver Operating Characteristics Curve and Confusion Matrix to show the result of the Multinomial Naive Bayes classifier. As shown in the Figure5, we could find that the ROC curve gives an area of 0.86. According to the heatmap of the confusion matrix, we can see that the precisions, recalls, and f1_scores of the 3 classes which -1 stands for negative, 0 stands for neural and 1 stands for positive. The overall accuracy of this classifier is 68.82% which is a little lower than what we expected. The whole time of our Naive Bayes classifier is 3.5s on the testset.



(a) heat map of Naive Bayes confusion matrix



(b) AUC-ROC curve

	precision	recall	f1-score	support
-1.0	0.4874	0.7549	0.5924	317118
0.0	0.1810	0.5856	0.2765	201468
1.0	0.9642	0.6880	0.8030	2179606
accuracy			0.6882	2698192
macro avg	0.5442	0.6762	0.5573	2698192
weighted avg	0.8497	0.6882	0.7390	2698192

(c) accuracy and confusion matrix of Naive Bayes

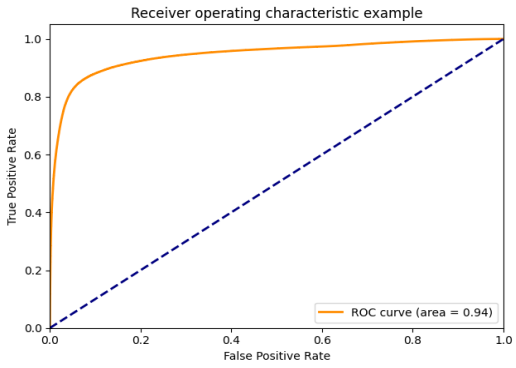
Figure 5: Naive Bayes evaluation performance

8.2 Support Vector Machine

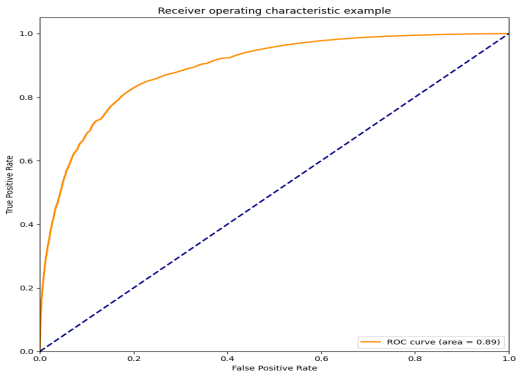
In this project, we realized two different SVM classifiers with sklearn library. Our SVM classifiers have different kernels, one is linear kernel and the other is RBF kernel. According to the results shown in Figure 6, we can see that the SVM model using linear kernel has a better performance on both accuracy and AUC score. The SVM classifier with linear model achieved an accuracy of 79.775% with an area of 0.94 in the ROC curve, while the RBF kernel got 68.983% in accuracy and an area of 0.89 in the ROC curve. The time cost of linear SVM kernel is 257.59s while the time cost of the RBF SVM classifier is 10 times longer which reached 2582.99s. We forgot to plot the confusion matrix of the RBF SVM classifier, we will add it in the final report.

8.3 Long Short-Term Memory

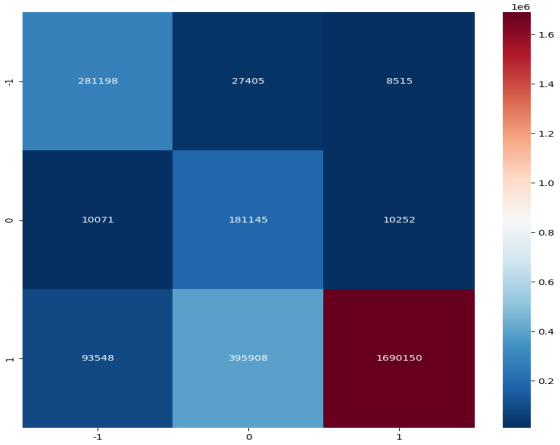
As with the previous two models, we used the same evaluation criteria to evaluate our LSTM model. According to the Figure7, we find that the final classification accuracy of our model is 73.4% with an area of 0.91 in ROC curve. Here we use 0,1,2 to classify the neutral, positive and negative comments, so it looks different from the previous two model's confusion matrices. We can also see that the precisions, recalls, and f1_scores of the 3 classes. While training the LSTM model, it took about 140s averagely to train each Epoch, and we set 20 Epoch to train it, so the training progress totally took about 2800s.



(a) AUC-ROC curve-linear kernel



(b) AUC-ROC curve-rbf kernel



(c) heat map of linear SVM confusion matrix

```
CalibratedClassifierCV(base_estimator=LinearSVC(C=0.5), cv=5)
training time: 257.59s
Model accuracy = 79.775%
              precision    recall  f1-score   support

        -1.0     0.7307    0.8867    0.8012    317118
         0.0     0.2997    0.8991    0.4495    201468
         1.0     0.9890    0.7754    0.8693   2179606

    accuracy                         0.7978   2698192
   macro avg     0.6731    0.8538    0.7067   2698192
weighted avg     0.9072    0.7978    0.8300   2698192
```

(d) accuracy and confusion matrix of Linear SVM

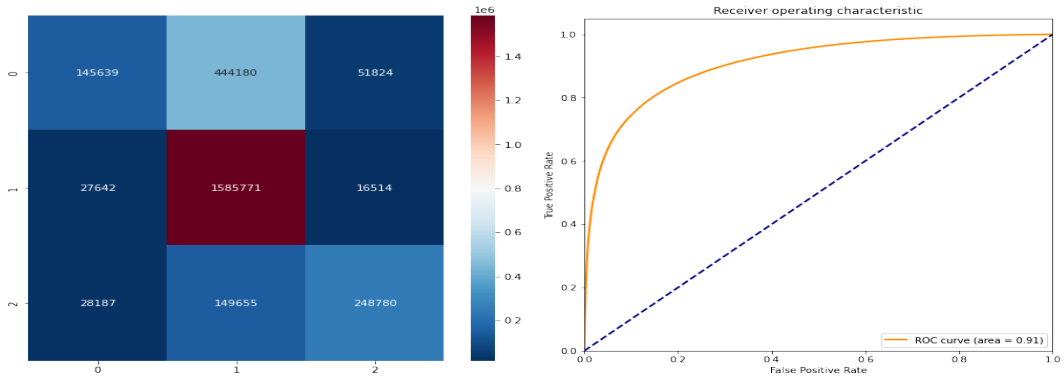
```
CalibratedClassifierCV(base_estimator=SVC(C=10, gamma=0.1), cv=5)
training time: 2582.99s
Model accuracy = 68.903%
              precision    recall  f1-score   support

        -1.0     0.5561    0.6897    0.6157     15872
         0.0     0.1765    0.6531    0.2779     10006
         1.0     0.9651    0.6922    0.8062    109032

    accuracy                         0.6890    134910
   macro avg     0.5659    0.6783    0.5666    134910
weighted avg     0.8585    0.6890    0.7446    134910
```

(e) accuracy and confusion matrix of rbf SVM

Figure 6: Accuracy and confusion matrix of SVM



(a) heat map of LSTM model confusion matrix

(b) AUC-ROC curve

	precision	recall	f1-score	support
0.0	0.227	0.723	0.345	201468
1.0	0.973	0.728	0.833	2179606
2.0	0.583	0.785	0.669	317118
accuracy			0.734	2698192
macro avg	0.594	0.745	0.616	2698192
weighted avg	0.871	0.734	0.777	2698192

(c) Accuracy of LSTM model

Figure 7: LSTM Model evaluation performance

9 Summary

9.1 Conclusion

In this project, we focus on learning and implementing how to apply machine learning algorithms to big data on media. Not only data pre-processing, but also how to design efficient machine learning algorithms to process data and perform sentiment analysis.

The purpose of this project is to find out which algorithms we used would perform better and produce the best result in sentiment analysis. According to the results we have got, the results are surprising. It's not the same as we thought, based on the results, we can finally conclude that **SVM with linear kernel model which we have created shows the best performance on sentiment analysis of our selected dataset**. We will then discuss the conclusion in four evaluation factors and give the possible reasons for conflicts with hypothesis.

9.1.1 Analysis on four evaluation factors

First is about classification accuracy, we find that the accuracy of SVM-linear model is 79.775%, which is larger than the Naive Bayesian, SVM with RBF kernel and LSTM model. Besides, for the AUC feature, the number of linear SVM is 0.94. And the results of the Naive Bayesian is 0.86, for RBF kernel SVMS is 0.86, while the result of LSTM is 0.91. And for confusion matrix feature, although some of the classification does not get the results, confusion matrix is the basis of AUC score, so the performance of AUC is consistent with that of confusion matrix.But for training time, we can easily find out the best performance model. Naive Bayesian cost quite a little time we finally get the result, while SVM with RBF kernel does not have a good performance. We consider the reason is that since SVM uses the Kernel Matrix of the data set to describe the similarity between samples, the number of matrix elements increases as

the data size increases. In this way, as the size of the data increases, the time it takes will also become longer. So, for the big data analysis, SVMs of linear kernel may cost quite a lot of time. And for LSTM, the time it cost is 140s for one epoch. Since here we run about 20 epochs for training, so the total time cost of LSTM is larger than that of the SVM-linear model.

Generally, the Linear SVM performs best on accuracy,AUC, cost a little more time than Naive Bayesian but be faster than SVMs with RBF and LSTM model. And its generalization performs good as well. So we can compare that SVM-linear model will be the best model for the selected dataset.

9.1.2 Reasons for inconsistency with hypothesis

As we mentioned before, our final conclusion is objective to the hypothesis. We finally derived four possible reasons from the aspects of data characteristics and algorithm construction.
1.Data set is more like linear classification.
We can find that the accuracy score of Linear SVM is the highest one among four models. Also, it is the only algorithm that use linear classification.Therefore, based on the above two basic facts, we boldly conclude that the characteristics of this data set are linearly separable, so this is also one of the important reasons why the Linear SVM performs optimally.
2.No sequential relationship among movie reviews.
Our dataset topic is about movie review which is a special type of text. As the description of LSTM model, it performs best in sentiment analysis in which sequential relationship is considered in long text. However, the text construction of movie reviews is not such complicated as other text. The emotional vocabulary in film reviews is very fixed and strong, and it can be classified without sequential relationship. Therefore, we believe that the performance of the LSTM model is not necessarily optimal based on the movie review data set.
3. Time limitation to find the best LSTM model.
In addition to some objective reasons, we believe that there are some subjective factors that can be improved. According to the loss function of our existing training model and the change trend in accuracy rate, we can clearly conclude that the existing LSTM model does not get the optimal model, so it may also be an important reason that the LSTM model is not better than the linear SVM in accuracy and AUC parameters.
4.Sequential iteration is quite expensive.
As defined by our research problem, the best model not only considers the accuracy and other evaluation parameters scores, but also considers the cost which refers to training time. According to the algorithm, we can know that LSTM is trained in sequential iterations, that is to say, in each epoch, the entire training set needs to be trained to obtain the optimal parameters in one single epoch and then iterate such process until reach the setting epoch number. This is undoubtedly an expensive process, especially for big data project. Therefore, if the accuracy and other parameters are not significantly better than those of other models, the high cost of the LSTM model is a shortcoming that is difficult to ignore. Even as stated in the third reason, the optimal model is finally found, but according to the existing results, the gap of Linear SVM and LSTM models in accuracy is small, but the time consumption gap is huge. Therefore, considering the starting point of training time, the expensive factor of sequential iteration cannot be avoided.

9.2 Discussion

In this section, we will mainly talk about whether our results is successful, analysis what kind of improvements can be done in the future work. explain the problems we met and then draw conclusions based on our research questions and hypotheses.

9.2.1 Improvement

Although we have successfully trained several models of the classifier, we can see from the accuracy of the model and the training time of the model that there is still room for improvement. In fact, due to time constraints, we had some ideas that could be used for improvement that were not put into practice.

For example, in the section on data pre-processing, in the section on tokenization, some abbreviations may be taken into account. Many people abbreviate "not" when writing comments, which can make it misleading to analyse the emotion of a statement after splitting the word. Beyond that, English words come in many forms. Perhaps we could also take these complex transformations into account when pre-processing. For example, Lemmatization and Stemming. We also try to improve the performance of the models we used.

In the process of training the LSTM model, we used a different number of Epochs to train the model, changing from 10 to 20. A model trained with a larger number of Epochs will give a higher accuracy, so perhaps if there is more time, we could try to increase the number of Epochs so that the model is not overfitted but also improves the accuracy and other evaluation features of the prediction. So, if we continue to enlarge the number of the epochs for LSTM model, the final best performing model might be the LSTM model rather than SVM-linear.

9.2.2 Problems

We also face several problems in our project process, first is time-consuming. Since all the models shall modify the features to find the best one, so we change the features at the same time on three computers, so that it will speed up. Also we face the problem of the labels in LSTM model. At first, our loss function performs quite wield which is approximately equal to 0. After searching the Keras official documents, it does exist problem if there are negative number as labels, so we finally transform the labels, and the problem becomes solved.

10 Responsibility

For the coding part of this project, Xin Sun did the code implementation of data preprocessing, training and testing the Naive Bayesian model as well as the SVM model. And we did the implementation of the LSTM model together. After the initial work on the code, we tested and trained the model together and made some changes based on the results of the code ran. As for the report, we wrote the Summary and Related work section together. Besides that, Xin Sun wrote the sections of Data Preprocessing and Result, Yuan Wo wrote the section of Research question, Hypothesis, Project Plan and Methodology, and Yuxin Meng wrote the rest sections.

References

[1] Lstm working process. [EB/OL]. http://www.open-open.com/lib/view/open1440843534638.html.

[2] Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. Sentiwordnet 3.0: an enhanced lexical resource for sentiment analysis and opinion mining. In *Lrec*, volume 10, pages 2200–2204, 2010.

[3] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[4] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. A neural probabilistic language model. *The journal of machine learning research*, 3:1137–1155, 2003.

[5] Erik Boiy, Pieter Hens, Koen Deschacht, and Marie-Francine Moens. Automatic sentiment analysis in on-line text. In *ELPUB*, pages 349–360, 2007.

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[7] Alex Graves. Supervised sequence labelling. In *Supervised sequence labelling with recurrent neural networks*, pages 5–13. Springer, 2012.

[8] Mohammad Sadegh Hajmohammadi and Roliana Ibrahim. A svm-based method for sentiment analysis in persian language. In *International conference on graphic and image processing (ICGIP 2012)*, volume 8768, page 876838. International Society for Optics and Photonics, 2013.

[9] Abdalraouf Hassan and Ausif Mahmood. Deep learning approach for sentiment analysis of short texts. In *2017 3rd international conference on control, automation and robotics (ICCAR)*, pages 705–710. IEEE, 2017.

[10] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[11] Thorsten Joachims. Text categorization with support vector machines: Learning with many relevant features. In *European conference on machine learning*, pages 137–142. Springer, 1998.

[12] Dan Li and Jiang Qian. Text sentiment analysis based on long short-term memory. In *2016 First IEEE International Conference on Computer Communication and the Internet (ICCCI)*, pages 471–475, 2016.

[13] Avinash Madasu and Vijjini Anvesh Rao. Sequential learning of convolutional features for effective text classification. *arXiv preprint arXiv:1909.00080*, 2019.

[14] J. McAuley and J. Leskovec. Web data: Amazon movie reviews. https://snap.stanford.edu/data/web-Movies.html, June 2013.

[15] Walaa Medhat, Ahmed Hassan, and Hoda Korashy. Sentiment analysis algorithms and applications: A survey. *Ain Shams engineering journal*, 5(4):1093–1113, 2014.

[16] Volodymyr Mnih, Nicolas Heess, Alex Graves, et al. Recurrent models of visual attention. In *Advances in neural information processing systems*, pages 2204–2212, 2014.

[17] MS Neethu and R Rajasree. Sentiment analysis in twitter using machine learning techniques. In *2013 Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT)*, pages 1–5. IEEE, 2013.

[18] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. Thumbs up? sentiment classification using machine learning techniques. *arXiv preprint cs/0205070*, 2002.

[19] Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*, 2018.

[20] Jürgen Schmidhuber. Deep learning in neural networks: An overview. *Neural networks*, 61:85–117, 2015.

[21] Chao Yang, Hefeng Zhang, Bin Jiang, and Keqin Li. Aspect-based sentiment analysis with alternating coattention networks. *Information Processing & Management*, 56(3):463–478, 2019.