International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878 (Online), Volume-11 Issue-2, July 2022

Sentiment Analysis for Amazon Product Reviews

Apoorva Verma, Chirag Rawat, Shilpy Gupta

Abstract*: Sentiment analysis is a classicfication process whereby machine learning techniques are applied on text-driven datasets in order to analyse the emotion / opinion expressed in a text, e.g. a message being positive or negative about a certain topic. The problem is to conduct a sentiment analysis (positive and negative sentiment) on online product reviews of Products (unlocked mobile phones) sold on Amazon.com. The trained model can be used to predict users' sentiment based on their online reviews. In this project, different machine learning algorithms are compared, trained and tested on a dataset containing 400000 reviews. The performance of three different algorithms were compared: Multinomial Naive Bayes (MNB), Logistic Regression and Long short-term memory network (LSTM). The Logistic Regression model resulted in the highest performance with Accuracy of 0.95 and AUC of 0.94. The dataset consists of 400 thousand reviews of products (unlocked mobile phones) sold on Amazon.com which is publicly available on Kaggle. Solution to the problem would be useful for a brand to gain a broad sense of user's' sentiment towards a product through online reviews Further study is needed to investigate if the classfication remains accurate when including more than two classes (e.g. Introducing a neutral class).

Keywords: Sentiment Analysis (Positive and Negative sentiment), Unlocked Mobile Phones, online product reviews, Amazon

I. INTRODUCTION

Online shopping has been growing for 20 years and many e-commerce websites such as Amazon, Flipkart, eBay, etc have been created to meet the increasing demand. Consequently, a specific product can be bought on several websites and the prices may vary. As customers usually want the best quality for the lowest price but can't directly check it, reviews from other customers seem to be the most reliable way to decide whether to buy the product or not. Therefore, sentiment analysis has proven essential to understand a product's popularity among the buyers all over the world. This approach has a wide range of applications. Businesses, for example, are always interested in public or consumer thoughts and feelings about their products and services. Before using a service or purchasing a product, potential customers want to know what other people think about it.

Manuscript received on 11 June 2022 | Revised Manuscript received on 02 July 2022 | Manuscript Accepted on 15 July 2022 | Manuscript published on 30 July 2022.

* Correspondence Author

Apoorva Verma*, Department of Computer Science, Galgotias University, Gautam Buddha Nagar (Uttar Pradesh), India. E-mail: apoorva92verma@gmail.com

Chirag Rawat, Department of Computer Science, Galgotias University, Gautam Buddha Nagar (Uttar Pradesh), India. E-mail: chirag.rawat1000@gmail.com

Mrs. Shilpy Gupta, Department of Computer Science, Galgotias University, Gautam Buddha Nagar (Uttar Pradesh), India. E-mail: shilpy.gupta@galgotiasuniversity.edu.in

Last but not least, researchers use this data to conduct in-depth analyses of market trends and customer sentiment, which could lead to improved decisions.

A. Related works

Multiple studies about sentiment analysis on Amazon.com reviews have been done [1]. These studies used conventional Machine Learning (ML) like Naive Bayesian (NB), Support Vector Machine (SVM), decision trees or logistic regression, which resulted in relatively good performances (accuracy > 0.90). A sentiment analysis of reviews of Amazon beauty products has been conducted in 2018 by a student from KTH [2] and he got accuracies that could reach more than 90% with the SVM and NB classifiers. He found that SVM was performing better than NB for a large amount of data. He also focused on summaries of the reviews which are more informative and got higher accuracy than with the complete reviews. Xing Fang and Justin Zahn analysed different categories of Amazon products (beauty, book, electronic, and home) [3] with 3 different classifiers: NB, SVM and Random Forest. They reached the conclusion that Random Forest usually provided them with more accurate results. They also found that SVM was performing better than NB for larger data sets Some work has also been done about binary classification with LSTM network.

- Zhenxiang Zhou and Lan Xu analysed the usefulness of Amazon food reviews [4] with LSTM and feed-forward neural (FFN) networks. The results have shown that LSTM outperformed FFN, and that the accuracy was quite good ('80%).
- Reviews of Amazon books have also been analysed, using LSTM algorithm in 2017 [5][6-8]. They compared two recurrent neural networks (RNN): Gated Recurrent Unit (GRU) and Bidirectional LSTM. The bag-of-word algorithm was used for feature extraction. With a data set of more than 210 000 reviews, they got the best accuracy with the LSTM algorithm (86%).

The aim of this project is to investigate if sentimental analysis is feasible for the classification of product reviews from Amazon.com. Therefore, we will compare the performance of different classification algorithms on the binary classification (positive vs. negative) of product reviews from Amazon.com. Thereby, we want to investigate whether the category of products the reviews come from influence the performance of this classification. Once found the best performing classifier, it will be applied on new Amazon.com datasets containing reviews of different product categories and these results will be compared.

II. LITERATURE SURVEY

This section contains the literature review of the methods that have been used in this research.

In this research, there are two main steps involved in text classification. First, we need to find a word embedding to convert text into numerical representations. Second, we fit the numerical representations of text to machine learning algorithms or deep learning architectures. The dataset consists of 400 thousand reviews of unlocked mobile phones sold on Amazon.com. The data was acquired in December, 2016. There are 69% reviews with positive sentiment (rating > 3), 23% reviews with negative sentiment (rating < 3), and 8% reviews with neutral sentiment (rating = 3). Apparently, there are more positive sentiments than negative sentiments in this dataset. One common approach of word embedding is frequency based embedding such as BoW model.

B. Bag Of Words

This sectionOne common approach of word embedding is frequency based embedding such as BoW model. BoW model learns a vocabulary list from a given corpus and represents each document based on some counting methods of words.

C. CountVectorizer

We have implemented CountVectorizer in sklearn to compute occurrence counting of words. Some words might appear frequently but have little meaningful information about the sentiment of a particular review.

D. TfidfVectorizer

We used tf-idf transform to scale down the impact of frequently appeared words in a given corpus. Thus, I have also implemented TfidfVectorizer in sklearn to compute tf-idf weighted counting.

E. Naive Bayes

We will use one of the Naive Bayes (NB) classifier for defining the model. Specifically, we will use MultinomialNB classifier.
Naive Bayes Model
This model applies Bayes theorem with a Naive assumption of no relationship between different features. According to Bayes theorem:
Posterior = likelihood * proposition/evidence or
P(A|B) = P(B|A) * P(A)/P(B)
For ex: In a deck of playing cards, a card is chosen. What is the probability of a card being queen given the card is a face card? This can be solved using Bayes theorem.
P(Queen given Face card) = P(Queen|Face) P(Face given Queen) = P(Face|Queen) = 1 P(Queen) = 4/52 = 1/13 P(Face) = 3/13 From Bayes theorem:
P(Queen|Face) = P(Face|Queen) P(Queen)/P(Face) = 1/3
For an input with several variables: P(y|x1, x2, … xn) = P(x1, x2, … xn|y)* P(y)/P(x1,x2, …xn) with Naive Bayes we assume x1, x2 … xn are independent of each other, i.e:
P(x1, x2, … xn|y) = P(x1|y) * P(x2|y) … * P(xn|y)
The assumption in distribution of P(xi|y) give rise to different NBM. Naive Bayes Model works particularly well with text classification and spam filtering. Advantages of working with NB algorithm are:

- Requires a small amount of training data to learn the parameters
- Can be trained relatively fast compared to sophisticated models

The main disadvantage of NB Algorithm is:

- It's a decent classifier but a bad estimator
- It works well with discrete values but won't work with continuous values (can't be used in a regression)

A. Logistic Regression

we make use of the Logistic Regression algorithm to build the model. It identifies the probability of occurrence of an event by fitting data to a logit function. The equation used in the algorithm is:
$\log (p\ 1-p) = \beta_0 + \beta(num)$
Here, If the log(p/(1-p)) is greater than zero, then the success ratio is every time appears to be greater than half of the 100 percent.
F1-Score = 2 * ($A*B\ A+B$)
F1-Score can be a defined as harmonic mean of A and B.

B. Long Short-Term Memory

LSTM is an updated version of Recurrent Neural Network to overcome the vanishing gradient problem. Below is the architecture of LSTM with an explanation. It has a memory cell at the top which helps to carry the information from a particular time instance to the next time instance in an efficient manner. So, it can able to remember a lot of information from previous states when compared to RNN and overcomes the vanishing gradient problem. Information might be added or removed from the memory cell with the help of valves. LSTM network is fed by input data from the current time instance and output of hidden layer from the previous time instance. These two data passes through various activation functions and valves in the network before reaching the output.

III. METHOD

C. Data acquistino

The dataset consists of 400 thousand reviews of unlocked mobile phones sold on Amazon.com. The data was acquired in December, 2016 by the crawlers build by PromptCloud and is available to public on Kaggle. The data set has the following fields.

- Product Title
- Brand
- Price
- Rating (1 to 5)
- Review text
- Number of people who found the review helpful

Below are some summary statistics about the data.

```
Summary statistics of numerical features :
              Price         Rating   Review Votes
count  407907.000000  413840.000000  401544.000000
mean      226.867155       3.819578       1.507237
std       273.006259       1.548216       9.163853
min         1.730000       1.000000       0.000000
25%        79.990000       3.000000       0.000000
50%       144.710000       5.000000       0.000000
75%       269.990000       5.000000       1.000000
max      2598.000000       5.000000     645.000000

Total number of reviews:  413840

Total number of brands:  385

Total number of unique products:  4410

Percentage of reviews with neutral sentiment : 7.68%

Percentage of reviews with positive sentiment : 68.86%

Percentage of reviews with negative sentiment : 23.45%
```

A.Data pre- processing

For illustrative purpose, we have considered data with positive sentiment (rating = 4, 5) and negative sentiment (rating = 1, 2) and worked on 10% of the training data due to limitation of computational resources. To prepare the data, reviews with ratings greater than 3 will be labeled and encoded as "1" (positive sentiment) and reviews with ratings less than 3 will be labeled and encoded as "0" (negative sentiment). Neutral reviews (rating=3) will not be used. Since it is a huge data set, the labeled data have been split into training set and validation set in 90/10 so that the model could be trained on more data to improve accuracy. Text preprocessing is needed to convert raw reviews into cleaned review. Necessary steps include conversion to lowercase, removal of non-characters, removal of stop words, removal of html tags. Depending on machine learning algorithms or deep architectures used, cleaned text data are further transformed to suitable numerical representations, such as 1D feature vector for supervised learning algorithms or 2D tensor for LSTM.

B. Feature extraction

One common approach of word embedding is frequency based embedding such as BoW model. BoW model learns a vocabulary list from a given corpus and represents each document based on some counting methods of words. We have implemented CountVectorizer in sklearn to compute occurrence counting of words. Some words might appear frequently but have little meaningful information about the sentiment of a particular review. Instead of using occurrence counting, we can use tf-idf transform to scale down the impact of frequently appeared words in a given corpus. Thus, we have also implemented TfidfVectorizer in sklearn to compute tf-idf weighted counting.

C. Classifiers

To investigate which machine learning modality performs best on the classification of Amazon.com reviews, three different machine learning modalities were trained.
•Multinomial naive Bayesian (MNB)
•Logistic Regression
•Long short-term Memory (LSTM) networks.
Since Naive Bayes and SVM are considered as conventional algorithms and are widely used in the field of sentimental analysis, we will use these classifiers as benchmarks. LSTM is a newer technique and is shown to have a high potential for a good performance in sentiment analysis. For the LSTM networks, We created our model with Keras library, which consists of 4 layers:
•Embedding: Reducing the size of inputs
•Spatial dropout: To prevent from overfitting
•LSTM: Long Short Term Memory layer, which is the RNN
•Dense: To convert LSTM outputs to binaries

D. Training / Testing process

To keep the computational expense limited, a randomly selected subset out of the training dataset, consisting 60.000 reviews, was extracted. From this 60.000 reviews, 40.000 were used to train the classifiers whereas the remaining 20.000 were used to test their performances.

E. Evaluating matrices

To decide which classification algorithm performed the most accurate on the test set, we used several performance metrics:

- Accuracy: it compares the predicted general sentiment (positive or negative) to the
- real one, which was determined based on the stars.
- AUC: The Area Under Curve (AUC) is a metric where the False Positive Rate (FPR) and True Positive Rate (TPR) are combined into one single metric. First, the FPR ad TPR are computed with many different thresholds for the classification algorithm. These FPRs and TPRs are parametrically plotted in a single graph, which results in the Receiver Operating Characteristic (ROC) curve. Finally, the metric we consider is the Area of this curve, which we call AUROC or AUC.
- Precision: this is the ration between True Positives and the sum of True Positives and False Positive reviews. It tells us how accurate we are about saying that a review is positive.
- Recall: this is the ration between True Positives and the sum of True Positives and False Negatives.
- F1-score: this is the harmonic mean of the precision and the recall

G.Evaluation Process

To decide which classification algorithm performed the most accurate on the test set, we used several performance metrics:

- Accuracy: it compares the predicted general sentiment (positive or negative) to the
- real one, which was determined based on the stars.
- AUC: The Area Under Curve (AUC) is a metric where the False Positive Rate (FPR) and True Positive Rate (TPR) are combined into one single metric. First, the FPR ad TPR are computed with many different thresholds for the classification algorithm. These FPRs and TPRs are parametrically plotted in a single graph, which results in the Receiver Operating Characteristic (ROC) curve. Finally, the metric we consider is the Area of this curve, which we call AUROC or AUC.
- Precision: this is the ration between True Positives and the sum of True Positives and False Positive reviews. It tells us how accurate we are about saying that a review is positive.
- Recall: this is the ration between True Positives and the sum of True Positives and False Negatives.
- F1-score: this is the harmonic mean of the precision and the recall

IV. RESULT

Model	Trained on 1% of training data	Trained on 10% of training
CountVectorizer + Multinomial Naive Bayes	88.35%	91.84%
TfidfVectorizer + Logistic Regression	89.32%	93.10%
TfidfVectorizer + Logistic Regression	91.59%	95.63%
(best parameter set by GridSearch)		
Word2Vec Embedding + Random Forest Classifier	84.79%	92.26%
Simple LSTM	90.61%	94.14%
Word2Vec Embedding + LSTM	88.35%	94.40%

V. DISCUSSION

To summarize our work, we selected 60000 random reviews between 4 million reviews, cleaned and tokenized them, created different models and selected the best. After this process, we evaluated them with the remaining 3.94 million reviews and scraped ~230000 real time reviews from Amazon.com to answer our question "Are sentiment analysis methods feasible for Amazon.com reviews?".

VI. TEST AND EVALUATION RESULTS

The results from Multinomial Naive Bayes (MNB) and Logistic Regression (LSVM) were also satisfying, however, since TF-IDF vectorisation limits the data and tokenisation is not efficient for those classifiers, tokenising the sentences and training with neural networks gave the best results. Here we can see that not only the classification method, but also feature extraction has an important role in the process. Different types of Neural Networks may give more accurate results; however, when we see previous works and researches, we can say that Long-Short Term Recurrent Neural Networks are working pretty much for sentiment classification. Since the results are very comparable.

```
Confusion Matrix :
 [[ 648  130]
 [  83 2228]]
```

```
Confusion Matrix :
 [[ 622  156]
 [  96 2215]]
```

```
Confusion Matrix :
 [[ 699   79]
 [  57 2254]]
```

Confusion Matrices for LSTM classifications on the evaluation data set.

(all ≥0.90 accuracy) on different datasets, containing randomly selected reviews (Kaggle dataset) and reviews from different product categories collected by searching the Amazon.com webpage for different searching terms, it is very unlikely that the performance of the LSTM depends strongly on the category of the product reviews.

VII. CONCLUSION

We have created word clouds for positive sentiment reviews and negative sentiment reviews of a selected brand, to get an intuition of words that frequently appear in different sentiments. For example, reviews with positive sentiment on Apple's products frequently contain words such as "good", "great", "phone", whereas reviews with negative sentiment frequently contain words such as "battery", "unlocked", "n't".

We have implemented Tfidf Vectorizer with Logistic Regression and investigated the following results.

- Top 10 features with smallest coefficients: ['not' 'return' 'disappointed' 'waste' 'horrible' 'worst' 'poor' 'slow' 'stopped' 'doesn'].
- Top 10 features with largest coefficients: ['great' 'love' 'excellent' 'perfect' 'good' 'easy' 'best' 'far' 'amazing' 'awesome']

This visualisation helps us to justify a bit about how logistic regression learn classifying positive and negative sentiment in this setting.

FUTURE STUDIES

We can further improve the final model of LSTM by using more training data (I have only use 10% of the data!). We can also use pre-trained word embedding such as GloVe, fastText which are trained on much larger data set from external resources. It should give better performance and save time for training our own word embedding. Besides, we can try different LSTM architectures to enhance performance, for instance a deeper network, more neurons, etc., but we should also consider the tradeoff between computational time and model complexity.

REFERENCES

1. Data Source from Kaggle https://www.kaggle.com/PromptCloudHQ/amazon-reviews-unlocked-mobile-phon es
2. " Working with text Data" from sklearn http://scikit-learn.org/stable/tutorial/text_analytics/working_with_text_data.html
3. "Using pre-trained word embeddings in a Keras model" from Keras Blog https://blog.keras.io/using-pre-trained-word-embeddings-in-a-keras-model.html
4. "Deep Learning with Word2Vec" from Gensim https://radimrehurek.com/gensim/models/word2vec.html
5. "Deep Learning, NLP, and Representations" http://colah.github.io/posts/2014-07-NLP-RNNs-Representations/)
6. " An Intuitive Understanding of Word Embeddings: From Count Vectors to Word2Vec" https://www.analyticsvidhya.com/blog/2017/06/word-embeddings-count-word2vee c/
7. "Sequence Classification with LSTM Recurrent Neural Networks in Python with Keras" https://machinelearningmastery.com/sequence-classification-lstm-recurrent-neura l-networks-python-keras/
8. "Embedding and Tokenizer in Keras" http://www.orbifold.net/default/2017/01/10/embedding-and-tokenizer-in-keras/

AUTHORS PROFILE

Apoorva Verma, Department of Computer Science Galgotias University, Gautam Buddha Nagar E-mail: apoorva92verma@gmail.com

Chirag Rawat, Department of Computer Science, Galgotias University, Gautam Buddha Nagar E-mail: chirag.rawat1000@gmail.com

Mrs. Shilpy Gupta, Department of Computer Science, Galgotias University Gautam Buddha Nagar E-mail: shilpy.gupta@galgotiasuniversity.edu.in

Retrieval Number: 100.1/ijrte.B70990711222
DOI: 10.35940/ijrte.B7099.0711222
Journal Website: www.ijrte.org

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)