Sentiment Analysis in Amazon Review Dataset

Parallel with PySpark

Tao Shan
Faculty of Mathematics
University of Waterloo
Waterloo, ON, Canada
t4shan@uwaterloo.ca

Peiyi Zheng
Faculty of Mathematics
University of Waterloo
Waterloo, ON, Canada
p2zheng@uwaterloo.ca

ABSTRACT

Sentiment analysis on a large scale of data which is a challenging problem to solve. Every day on Amazon, there are millions of people who buy goods and services from this platform.[1] The number of reviews should be potentially very large to analyze. Thus parallel systems such as Hadoop and Spark are reasonable solutions for processing the data. In this paper, our group aims to use the spark system to implement BERT, Glove and other common machine learning models using the amazon review dataset, including steps for Explanatory Data Analytics, Natural Language Processing and Modeling. We also perform some interesting data science on this particular dataset. We found that the BERT model achieves the best accuracy of 81%. The result shows a parallel machine learning system that can apply to a potentially large sentiment analysis dataset, which can apply to different organizations to understand customer satisfaction.

CCS CONCEPTS

• Artificial intelligence • Natural Language Processing • Parallel Computing

KEYWORDS

Hadoop/Spark Parallel System, Data Science, Visualization, Natural Language Processing, Machine Learning, Sentiment Analysis

1 Introduction

Sentiment analysis is a natural language processing task that involves the use of computational methods to identify and extract subjective information from text data. In the context of Amazon reviews, sentiment analysis can be used to determine the sentiment or opinion expressed in a review automatically and classify it as positive or negative. This information can be valuable for businesses, as it allows them to understand customer sentiment and make data-driven decisions about their products and services.

Nowadays, more and more data needs to be processed for online shopping. There are more than 29 million customers on Amazon.com and millions of items in their catalogs, and several other major retailers have comparable data sources. [1] Spark has the ability to handle large volumes of data parallelly, including a rich set of APIs for data manipulation, transformation, and analysis. So we want to produce the sentiment analysis system on Spark.

The dataset was collected on the Kaggle Platform through Amazon's API, with 983 thousand reviews and 1.53GB for the full dataset.[2] According to this dataset, our group wants to solve which algorithm has the best performance based on model performance and time efficiency, including Glove, BERT, and Logistic Regression. Bert is known as Bidirectional Encoder Representations from Transformers, developed by Google, which is a popular model in language modelling.[3] Glove, known as short for Global Vectors for Word Representation, was developed by researchers at Stanford, which is used for creating word vectors and effectively captures the semantic meaning of words.[4] We are using accuracy score and time execution to compare the model performance and efficiency.

2 Related Works

There have been promising results with different machine learning methods for sentiment analysis on Spark. In [5], authors implement algorithms by Apache Spark's Machine learning library, MLlib. The models that are chosen for sentiment analysis are Naive Bayes, Logistic Regression and Decision Trees, and it shows the best result of F-Measure with 0.725. The paper also shows preprocessing methods, including Unigrams, Bigrams, Pos Tags, and dealing with URLs.

Another paper [6], which uses the BERT algorithm, compares the accuracy with or without spark NLP. Spark NLP methods give much higher accuracy, improving from 0.8444 to 0.9187. Also, the time efficiency improves from 35 minutes to 9 minutes. The Spark NLP includes Document

Assembler(), Sentence Detector(), Tokenizer(), and BertEmeddings(). [6]

The paper[7] solves the sentiment analysis problem by the Glove algorithm, including word preprocessing in the Skip-Gram model, Continuous Bag of Words, and Glove model. The author explains the details of the glove model and implements a possible sentiment analysis solution by visualization.

These related works show possible methods for doing sentiment analysis with spark, including models, NLP and measurements, to help us identify the potential challenges and pitfalls.

3 Methodology

3.1 Spark

Spark is a fast and powerful open-source big data processing framework built around speed, ease of use, and sophisticated analytics. It was originally developed at UC Berkeley's AMPLab in 2009 and open-sourced in 2010. In memory, Spark can run programs 100x faster than Hadoop MapReduce or on disk 10x faster. [8] The spark, which offers API in python for our sentiment analysis, helps us process the data with a fast performance

3.2 MLlib

Based on Apache Spark, Spark MLlib provides machine learning algorithms and utilities. For sentiment analysis on big data, Spark MLlib is an ideal tool for scaling machine learning algorithms to large datasets. Our group applies machine learning models on MLlib packages to fit with the large data size.

3.3 Model Selection

3.3.1 BERT

Bidirectional Encoder Representations from Transformers (BERT) is a model for pre-training deep bidirectional representations from the unlabeled text by jointly conditioning on left and right context in all layers. [9] The BERT model is pre-trained on a large corpus of the text of 2500M words, [10] which learn from a massive word database. BERT model including steps for pre-training and next sentence prediction. During the pre-training steps, the data needed to be separated into tokens, divided the tokens into segments, randomly blanked out some of the tokens randomly, then trained the model. The pre-training step allows the model to learn a general-purpose language representation that can be fine-tuned for specific tasks. After the BERT model finishes pre-training, the model also includes next-sentence prediction, training to predict whether two sentences are related to each other or not.

3.3.2 GLOVE

The glove model is a type of word embedding model that is trained on a large corpus of text data. Using this model, each word is represented as a high-dimensional vector such that semantically similar words are also represented in a similar way in terms of their vectors. In this way, the model can capture the semantic relationship between words in a way that can be used in a variety of natural language processing applications. [12]

3.3.3 Random Forest

Random forest uses random spaces and bagging to train on multiple decision trees. The model creates multiple decision trees during training and then uses the majority class predicted by these trees to make a final prediction. [14]

3.3.4 Naive Bayes

Naive Bayes is based on using probabilities to make predictions. In sentiment analysis, Naive Bayes calculates the probabilities of different words occurring in positive and negative sentences. The Naive Bayes classifier assumes that the classes in the data are categorical, and it uses the training data to learn the probabilities of different values occurring in each class. When given new data, it uses these probabilities to make predictions about which class the data belongs to. [13]

3.3.5 Logistic Regression

Logistic Regression finds the non-linear relationship between variables and predictors. The model takes a vector of variables as input, evaluates the coefficients or weights for each variable, and then uses these coefficients to predict the class of the given product review as a word vector. [13]

3.4 Natural Language Processing

There are multiple ways to do NLP analysis, as well as to process text data and do NLP analysis. From a machine learning perspective, there are five main steps, namely, Corpus Understanding, Tokenization, Stopword removal, Stemming, and Numerical Formation.[5]

Corpus A corpus is a large, structured collection of machine-readable texts that are produced in natural communication environments. In natural language

processing, corpora contain text and speech data that can be used to train artificial intelligence and machine learning systems. For a corpus to meet the training requirements, there are several criteria. Firstly, we need the corpus to be as large as possible. Large specialized datasets are critical for training algorithms designed to perform sentiment analysis. When it comes to the data within the corpus, high quality is critical. Since corpora require large amounts of data, even minor errors in the training data can lead to massive errors in the output of the machine learning system. Moreover, data cleaning is also essential for creating and maintaining a high-quality corpus. Data cleaning identifies and eliminates any errors or duplicate data, creating a more reliable corpus for NLP. Lastly, a high-quality corpus is a balanced corpus. While it is tempting to populate a corpus with everything available, failing to streamline and structure the data collection process can unbalance the relevance of the dataset.[11]

Tokenization The text sentence is sliced into sub-units, then the subunits are numericalized (mapped into vectors), then these vectors are input to the model for encoding, and finally output to the downstream task for further results. In addition, we do not numerate the input sentence or word directly. We need to slice it into a finite number of subunits and then numerate these subunits. This process of cutting the original text into subunits is called Tokenization.

Stopword removal The marker field contains very common words such as ‘this', 'that', 'is', 'the', etc., which are called stop words and have limited value for the analysis. Using these words in the analysis will add computational overhead without adding much value or insight. Therefore, we always think about removing these stop words from tokens. In PySpark, we use StopWordsRemover to remove these stop words.[11]

4 Results

In this session, our group presents the steps and results for model comparison of BERT, GLOVE and other models with Spark. We also provide interesting data science to generate insights from the current dataset, including data exploration, interactive visualization, Natural Language Processing, and detailed solutions compared with prior works and challenges. For all the steps, we are using Spark to parallelize the process in order to fit with the potentially large dataset.

4.1 Dataset

Our group uses the Amazon review dataset [2] that is extracted from Amazon product review API [10] by the Kaggle Platform, including nearly 980 thousand records. We split the labels with 58.5% of positive sentiment and 41.5% for negative sentiment.

4.2 Design and Goal of the system

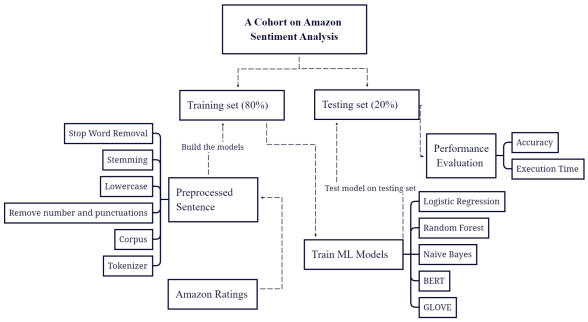

Figure 1: **URL diagram for workflow**

The design of the system shows in our URL diagram (Figure 1), including details for data preprocessing, model choices, split train/test data, and performance evaluation. Our project goal is to find the best model for sentiment analysis that parallels with Spark. Also, we want to generate insights and analytics from the Amazon review dataset.

4.3 Data Exploration

Our group is using word cloud, histogram and pie chart to explain the insights from our data. The data are preprocessed by PySpark, then converted to the format to fit with the python matplotlib library.

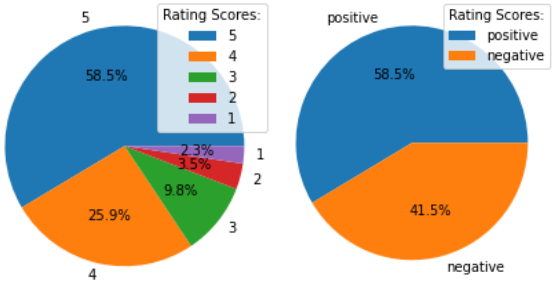

Figure 2: **pie charts for original label and sentiment label.**

The pie charts in figure 2 show the original rating labels and the sentiment labels by combining the labels that are not satisfied with the product. 58.5% of the records show

satisfaction fully, and 41.5% of records show somewhat/complete dissatisfaction.

The word cloud is a visual representation of data, which shows the common words by larger size in the plot. Word cloud visualizes the important words that benefit further analysis.

Figure 3: **word cloud for positive sentiments**

Figure 4: **word cloud for positive sentiments**

From Figure 3, the most popular positive sentiments are book, love and read. From Figure 4, the most popular words for negative sentiment are book, story and read. We found there are many common words between positive sentiment and negative sentiment. Furthermore, the words such as book, character, author, and read show the dataset is mostly the reviews for books or online readings. So the plots help to understand what kind of data is within our dataset.

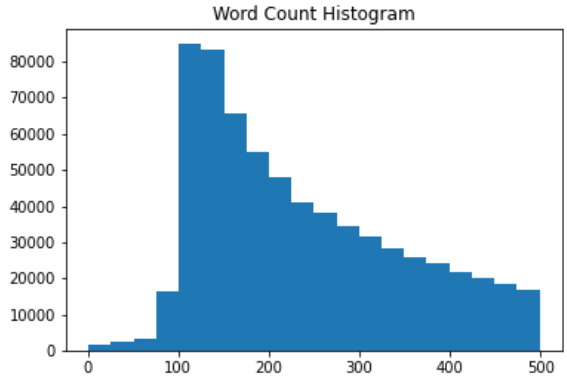

Figure 5: **word count for reviews**

The histogram (Figure 5) shows most reviews have words between 100 and 200, which shows a large vocabulary size for each record.

4.4 Interactive Plot

(the plot shown in interactive_plot.html and Interactive Plot.ipynb)

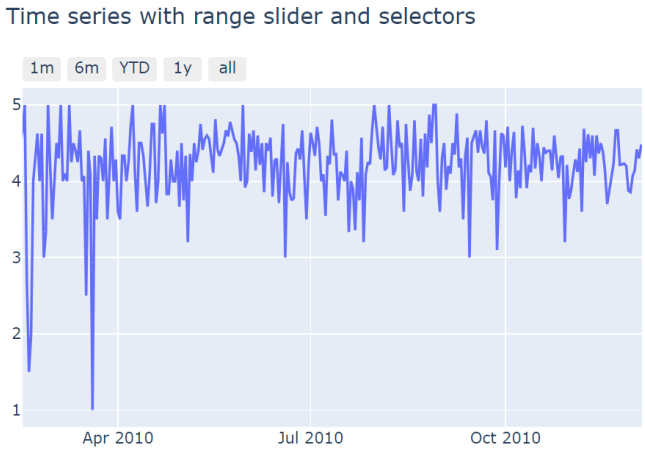

Figure 6: **interactive plot for time series data**

The purpose of this interactive plot (Figure 6) is to visualize the data of average rating in different time series. It allows users to choose a different range of time by clicking the range within the plot and shows the data monthly, semi-annually and annually. This plot helps the user understand the trend for the user's rating that shows the user's sentiment in different arrangements of the time.

4.5 Model Performance

From Table 1, we can tell Naive Bayes achieves the highest accuracy with One-Hot word encoding, while the accuracy of logistic regression and the random forest is 62.56% and 55.05% respectively.

Model	Logistic	Naive Bayes	Random Forest
Accuracy	62.56%	72.35%	55.05%

Table1. Result for standard ML models

From Table 2, we find that the classifier with Bert word embedding performed fairly well with a promising result in all the criteria compared with classic machine learning models.

Model	Glove	Bert
Accuracy	70%	**81%**
Precision	65%	**79%**
Recall	78%	**81%**
F1-score	64%	**79%**

Table2. Result for Word Embedding models

5 Discussion

5.1 Interpret the result

Due to the nature of our dataset, the accuracy of our model cannot reach the same level as mentioned in other papers. However, we can still say that the Bert algorithm has very stable and reliable performance in the task of natural language processing. We also found that Naive Bayes has satisfactory prediction accuracy and a surprising training speed. The random forest has a good performance in most of the machine learning tasks, but it does not seem to be suitable for our NLP analysis. For the data with different values of attributes, the attributes with more divided values will have a greater impact on random forest. So the random forest output on our data is not credible for attribute weights.

5.2 Discuss the implication

With MLib, Spark SQL and Spark NLP, we can easily process and analyze large-scale text data, and build and train models in parallel. The underlying RDD abstraction mechanism used by Spark builds the core data structure of the entire Spark ecosystem. For example, in our project, we first create an RDD with data files, then convert the RDD into a SchemaRDD and query it with Spark SQL. The result is then delivered to MLlib and Spark NLP. Depending on the task requirements, we can also continue to insert the resulting RDD into Spark streaming.

5.3 Limitation

Sentiment analysis is an extremely difficult task, even for humans. On average, inter-annotator agreement (a measure of the degree to which two human taggers can make the same annotation decision) is very low in sentiment analysis. Because machines learn from labeled data, sentiment analysis classifiers may not be as accurate as other types of classifiers.

However, in the dataset we are dealing with, there are no manually labeled emotional expressions. We are tagging users' emotions based on their rating of the item. However, such an overly absolute dichotomy might cause some errors.

There is more information available to us in the Amazon reviews: Kindle Store Category dataset. For example, for the helpfulness rating of the review, we can filter some of the reviews with low helpfulness ratings in the subsequent study to improve the accuracy of the prediction. At the same time, we can also add other categories of product reviews together for sentimental analysis. Moreover, we can do more experiments to test more hyper-parameters to make our model perform better.

6 Conclusion

We presented results for sentiment analysis on Amazon Review. As part of our work, we focus on three classic Machine Learning models and the two-word embedding model's performance, including data exploration, data preprocessing, and model evaluation. As a result of our analysis, we conclude that the BERT model performs best in Amazon sentiment analysis, but its computational cost is high. Naive Bayes model has a lower performance than BERT, but it still has a highly effective performance and a low computational cost.

In future works, we will try more models, such as RNN and LSTM. Moreover, we may compare the differences between different data preprocessing methods, such as the count tokenizer and TF-IDF's tokenizer. Different tokenizers may generate different vectors of word tokens and present different results. Moreover, we can apply grid search to test

more hyper-parameters to make our models achieve better performance.

REFERENCES

[1] G. Linden, B. Smith and J. York, "Amazon.com recommendations: item-to-item collaborative filtering," in IEEE Internet Computing, vol. 7, no. 1, pp. 76-80, Jan.-Feb. 2003, doi: 10.1109/MIC.2003.1167344.

[2] Amazon reviews: Kindle Store Category | Kaggle

[3] Hu Xu, Bing Liu, Lei Shu, and Philip Yu. 2019. BERT Post-Training for Review Reading Comprehension and Aspect-based Sentiment Analysis. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 2324–2335, Minneapolis, Minnesota. Association for Computational Linguistics.

[4] Y. Sharma, G. Agrawal, P. Jain and T. Kumar, "Vector representation of words for sentiment analysis using GloVe," 2017 International Conference on Intelligent Communication and Computational Techniques (ICCT), 2017, pp. 279-284, doi: 10.1109/INTELCCT.2017.8324059.

[5] Baltas, A., Kanavos, A., Tsakalidis, A.K. (2017). An Apache Spark Implementation for Sentiment Analysis on Twitter Data. In: Sellis, T., Oikonomou, K. (eds) Algorithmic Aspects of Cloud Computing. ALGOCLOUD 2016. Lecture Notes in Computer Science(), vol 10230. Springer, Cham. https://doi.org/10.1007/978-3-319-57045-7_2

[6] Kuncahyo Setyo Nugroho, Anantha Yullian Sukmadewa, and Novanto Yudistira. 2021. Large-Scale News Classification using BERT Language Model: Spark NLP Approach. In 6th International Conference on Sustainable Information Engineering and Technology 2021 (SIET '21). Association for Computing Machinery, New York, NY, USA, 240–246. https://doi.org/10.1145/3479645.3479658

[7] Y. Sharma, G. Agrawal, P. Jain and T. Kumar, "Vector representation of words for sentiment analysis using GloVe," 2017 International Conference on Intelligent Communication and Computational Techniques (ICCT), 2017, pp. 279-284, doi: 10.1109/INTELCCT.2017.8324059.

[8] Zhuolin Qiu, Bin Wu, Bai Wang, Le Yu Proceedings of the 3rd International Workshop on Big Data, Streams and Heterogeneous Source Mining: Algorithms, Systems, Programming Models and Applications, PMLR 36:17-28, 2014.

[9] Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding.

[10] http://jmcauley.ucsd.edu/data/amazon/

[11] https://hypersense.subex.com/aiglossary/corpus/

[12] Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global Vectors for Word Representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

[13] A. Prabhat and V. Khullar, "Sentiment classification on big data using Naïve bayes and logistic regression," 2017 International Conference on Computer Communication and Informatics (ICCCI), 2017, pp. 1-5, doi: 10.1109/ICCCI.2017.8117734.

[14] Yassine Al Amrani, Mohamed Lazaar, Kamal Eddine El Kadiri, Random Forest and Support Vector Machine based Hybrid Approach to Sentiment Analysis, Procedia Computer Science, Volume 127, 2018, Pages 511-520, ISSN 1877-0509, https://doi.org/10.1016/j.procs.2018.01.150.