Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 141 (2018) 183–189

Procedia
Computer Science

www.elsevier.com/locate/procedia

The 9th International Conference on Emerging Ubiquitous Systems and Pervasive Networks (EUSPN 2018)

A Large-Scale Sentiment Data Classification for Online Reviews Under Apache Spark

Samar Al-Saqqa[a,b,]*, Ghazi Al-Naymat[a], Arafat Awajan[a]

[a]*Princess Sumaya University for Technology, Amman, Jordan*
[b]*The University of Jordan, Amman, Jordan*

Abstract

Sentiment Analysis of large-scale data has become increasingly important and has attracted many researchers, urging them to use new platforms and tools that can handle large volumes of data. In this paper, we present new evaluation experiments of sentiment analysis for a large-scale dataset of online customer's reviews under Apache Spark data Processing System. Apache Spark's scalable machine learning library (MLlib) is used and three classification techniques from the library are applied; Naïve Bayes, Support vector machine, and logistic regression. The results are evaluated using the accuracy metric. Experimental results show that Support vector machine classifier outperforms Naïve Bayes and logistic regression classifiers.

Keywords: Big Data; Apache spark; Sentiment; MLlib; Machine Learning.

1. Introduction

Online customer reviews are increasingly available on a wide range of products and service and analyzing the sentiment of customers reviews has become very beneficial in business, where businesses can track positive and negative brand reviews that help them measure their overall performance and can play a key role in measuring sales and improving business marketing strategies as well. Customer's reviews are one of the large amounts of sized data,

* Corresponding author. Tel.+962795823983.
E-mail address: s.alsaqqa@ju.edu.jo

10.1016/j.procs.2018.10.166

because it contains millions of reviews from various websites that are increasing every day. This large volume of data that is increasing every second is known as big data, it includes a huge amount of structured, unstructured, semi-structured data [1]. This growth in data size and data structure diversity has become a challenge to traditional relational databases and enterprise data warehouses since it cannot be easily analyzed and requires more time and complex resources to be processed [2].

There are three main features of big data referred to as 3V'S; Velocity, Volume, and Variety. The feature "Volume" refers to the amount of large data generated from many different sources in a period of time. Velocity term is related to the speed at which the data is generated or analyzed. The different sources and types of data both structured and unstructured such as text, images, audio and video are known as the Variety [3]. Big data needs a powerful machine learning tools, strategies, and environments to be analyzed properly. Not all machine learning tools, such as R and Weka are capable of analyzing large volumes of data because data is too large to fit into the main memory of a single computer. These traditional tools cannot analyze a large amount of data, so to handle the large volume of data a new big data processing platform such as Apache Hadoop and Apache Spark are designed, these tools implemented the machine learning algorithms to achieve high efficiency [4], [5].

Apache Spark, developed at the University of California, Berkeley's in 2009, is an open-source processing framework. It has become one of the keys for large-scale data distributed processing and analytics frameworks in the world, it achieves high performance for both batch and streaming data and it has easy-to-use APIs for operating on large datasets. Spark can be 100x faster than Hadoop for large-scale data processing by exploiting in memory computing and other optimizations [6], [7]. MLlib is Apache Spark's scalable machine learning library built on top of Spark to deliver both high quality and high speed. MLlib can be used with Java, Scala, and Python, so that you can include it in complete workflows [8].

Sentiment analysis is considered a classification problem and can be solved using the machine learning approach [9], [10]. In our work, we used Spark's MLlib which is one of machine learning libraries that could be used for large-scale data classification, and since Spark's MLlib is a new library created in 2014, little research was done using Spark MLlib. According to researchers' best knowledge, a limited number of studies have been conducted to analyze the sentiment for large-scale data using Spark's MLlib, so further experimental work is required in this area. This research aims to provide new experiments of sentiment classification on large-scale data using the Spark's MLlib by applying different MLlib classification algorithms and evaluating their performance.

The rest of the paper is organized as follows; section two provides related work, section three presents the proposed approach, section four shows the experimental results, and section five provides the conclusion in addition to some insights toward potential future work.

2. Related Work

Sentiment analysis is one of the most active research areas that researchers focused on during the recent years, many researchers have used many different methods and algorithms for performing sentiment analysis. Nabil et al. [11] interested in testing the performance of different machine learning algorithms on a dataset of more than 10,000 Arabic tweets. They proposed to classify the texts into four categories: objective, subjective negative, subjective positive and subjective mixed. Among the experimented algorithms including (SVM, MBN, BNB, KNN, and stochastic gradient descent), they found that SVM is the best classifier. Duwairi and Qarqaz [12], applied three classifiers on a dataset of tweets. They used NB, SVM and KNN classifiers and compared the results of these three classifiers. Precision and recall rates indicate that SVM and KNN classifiers perform better than NB. Customer reviews are used by researchers in many different areas of interest. In [13], the authors proposed a classification and sentiment analysis system for Amazon reviews, the system classifies reviews into service, product, and feature based reviews and determine the sentiment for each review. [14] proposed an improved Naïve Bayes algorithm to be used for the sentiment analysis of restaurant reviews based on senti-lexicon. Nazlia et al. [15] proposed to study different machine learning classifiers framework mainly with the problem of subjectivity and sentiment analysis for Arabic customer reviews. They adopted three classifiers as base-classifiers (Naive Bayes, Rocchio classifier and support vector machines). A comparative study of the fixed combination and meta-classifier combination has been done. The experimental results show that the ensemble of the classification algorithms with meta-learner improves the classification effectiveness.

There is a lot of research conducted using Hadoop Map-Reduce for sentiment analysis [16],[17]. However, some research is conducted using Apache Spark framework for sentiment analysis. Baltas et al in [18] implemented a tool of sentiment analysis of Twitter data which used the Spark's machine learning library (MLlib) for classification. Three classifiers were used, decision tree, Naïve Bayes and logistic regression, binary and ternary classifications were examined on real data from Twitter. The data preprocessing is handled to improve the performance. The system is evaluated on different dataset sizes and different features. Authors indicated that the results of Naïve Bayes were better than the other classifiers results and the performance of classifiers could be affected by the size of the dataset.

In [19] Apache spark processing system is used to identify malicious users and analyze their behavior to enhance the accuracy of trust. The public trust standards are combined and a dictionary of malicious words is created to detect and analyze malicious users on more precise trust process, and by using a spark, they obtain the distributed environment and speed. In [20] the accuracy of different apache spark classifiers is evaluated for multi classes classification the performance is compared using different dataset sizes and different number of n-gram, the results revealed that logistic regression classifier has achieved higher accuracy compared with other classifiers and the classification accuracy increases when using combination of n-gram, and increasing the size of the data set has a positive impact on overall accuracy.

3. Proposed Approach

The main stages of the approach are presented in figure 1. The approach starting with the data preprocessing and feature extraction, then applying the machine learning classifiers: Naïve Bayes, Support vector machine and logistic regression separately under Spark environment. Finally, the results are evaluated using the accuracy metric.

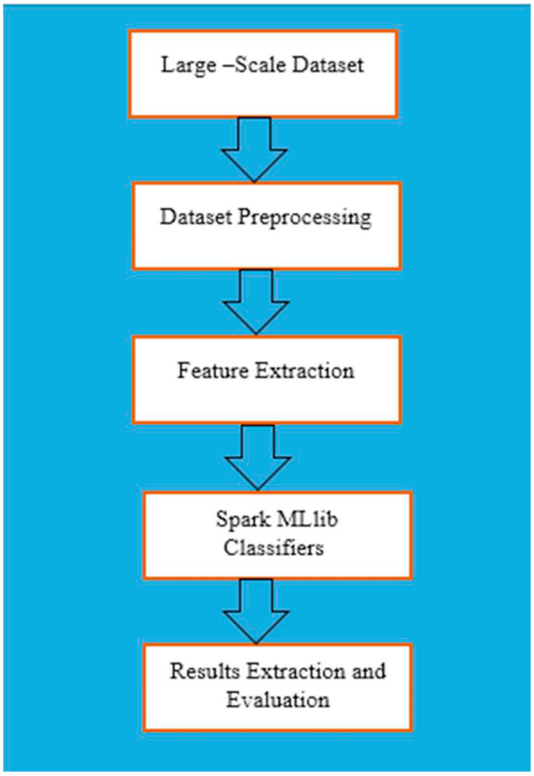


Fig 1. The main phases of the proposed approach

3.1. Dataset

The data set used for experiments is the Amazon review polarity dataset. It is constructed by Xiang Zhang [21] from the Amazon reviews dataset that consists of reviews from Amazon, which spans 18 years, including around 35 million reviews up to March 2013. The Amazon reviews polarity dataset includes a class index, review title and review text. It contains 1,800,000 training samples and 200,000 testing samples in each class, as shown in table 1. The histogram for reviews text length for each class is represented in figure 2.

Table 1: Amazon Review Dataset

	No. of Positive Reviews	No. of Negative Reviews
Training Sample	1,800,000	1,800,000
Testing Sample	200,000	200,000
Total	4,000,000	

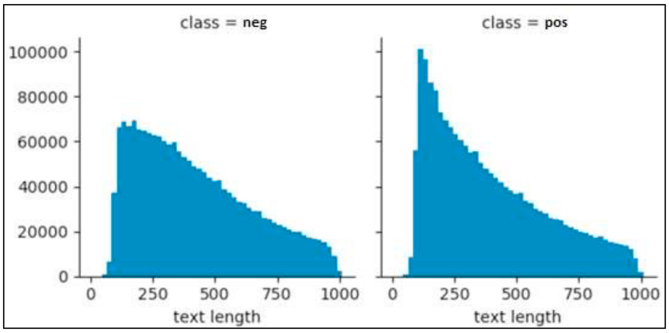


Fig. 2. The histogram represents the text length of reviews for each class.

3.2. Data Preprocessing

The preprocessing is applied to the dataset before passing to the classifier in order to cleanse and prepare it for classification, it involves:

- Removing null reviews: this includes removing the review that contains a null value, and after counting the number of reviews that contain empty values, the number of empty reviews is two.
- Tokenization: In this phase, the text is divided into multiple tokens based on separator characters such as white space, comma, tab, etc. The following is an example of tokenization:

Before Tokenization: ' *This sound track was beautiful! It paints the senery in your mind so well I would recomend it even to people who hate vid. game music! I have played the game Chrono Cross but out of all of the games I have ever played it has the best music! It backs away from crude keyboarding and takes a fresher step with grate guitars and soulful orchestras. It would impress anyone who cares to listen* '

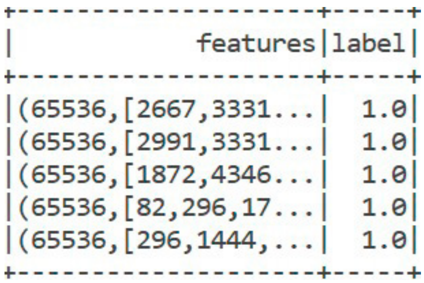
After:*' This', 'sound', 'track', 'was', 'beautiful', 'It', 'paints', 'the', 'senery', 'in', 'your', 'mind', 'so', 'well', 'I', 'would', 'recomend', 'it', 'even', 'to', 'people', 'who', 'hate', 'vid', 'game', 'music', 'I', 'have', 'played', 'the', 'game', 'Chrono', 'Cross', 'but', 'out', 'of', 'all', 'of', 'the', 'games', 'I', 'have', 'ever', 'played', 'it', 'has', 'the', 'best', 'music', 'It', 'backs', 'away', 'from', 'crude', 'keyboarding', 'and', 'takes', 'a', 'fresher', 'step', 'with', 'grate', 'guitars', 'and', 'soulful', 'orchestras', 'It', 'would', 'impress', 'anyone', 'who', 'cares', 'to', 'listen'*

- Noise removal: this step encompasses cleaning the text from some irrelevant information which may decrease the performance of the classifier such as numbers, punctuation marks, URL links, and special characters.

- Stop-words removal: stop words is common used words and which have little information content such as conjunctions, prepositions.

3.3. Feature extraction

Before using the data by machine learning classifier we need to represent the text in a format that suitable for the classifier to deal with it, so the text is converted into a feature vector. The Term Frequency-Inverse Document Frequency (TF/IDF) is used, where term frequency (TF) is the number of times that a word or term occurs in a text or review and the inverse document frequency (IDF) is a measure of how much information the word provides whether the term is common or rare across all documents. Figure 3 represents a sample of feature extraction output.



```
+--------------------+-----+
|            features|label|
+--------------------+-----+
|(65536,[2667,3331...|  1.0|
|(65536,[2991,3331...|  1.0|
|(65536,[1872,4346...|  1.0|
|(65536,[82,296,17...|  1.0|
|(65536,[296,1444,...|  1.0|
+--------------------+-----+
```

Fig. 3. A sample of output after feature extraction

3.4. Spark's MLlib Classifiers

Three Machine learning classifiers are applied; Naïve Bayes (NB), Support vector machine(SVM) and logistic regression.

- Naive Bayes(NB): it is a simple algorithm based on probabilistic Bayes' theorem that is described as the following equation:

$$\mathrm{P(A \mid B)} = \frac{\mathrm{P(B|A)P(A)}}{\mathrm{P(B)}} \tag{1}$$

where A and B are events and $P(B) \neq 0$. $P(A|B)$ is a probability (conditional probability) of the occurrence of event A given the event B is true. $P(B|A)$ is a probability of the occurrence of event B given the event A is true. $P(A)$ and $P(B)$ are the probabilities of the occurrence of event A and B respectively. NB constructs the model by adjusting the distribution of the number for each feature [22].

- Support Vector Machine(SVM): it is a supervised machine learning used generally in classification problems. In the SVM, each data item is considered as a point in n-dimensional space (where n is the number of features) where the value of a particular coordinate of each point represents a specific feature. The classification is performed by finding the hyperplane that maximizes the margin between the two classes and the vectors that define the hyperplane are the support vectors as shown in figure 4. The SVM is a powerful classifier which has been used successfully in many text classification problems. Text documents are represented as vectors and the model is built based on the extracted features from the training dataset, it tries to find a hyperplane represented by vectors that split the positive and negative training vectors of documents with maximum margin [23].
- Logistic regression: it is used to predict a binary response based on one or more independent variables or features. The two possible dependent variable values are often labeled as "0" and "1", so the result of logistic regression is the probability that the given input point belongs to a certain class. For example, if we have only two classes "0" and "1", and the probability *P (0)* is the probability that a certain data point belongs to the '0'class. Of course *P (1) =1-P (0),* where *P (1)* is the probability that a certain data point belongs to the '1'class. Thus, the output of logistic regression always lies in [0, 1].

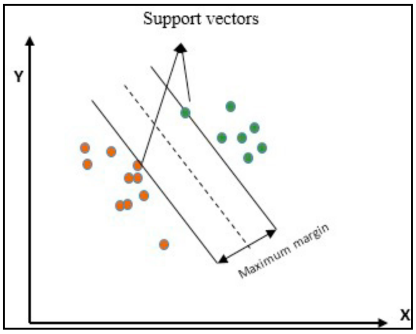


Fig. 4: Support vector machine

4. Experimental Results

The purpose of the experiments is to compare the performance of three MLlib's classifiers; Naive Bayes, support vector machine and logistic regression. To measure the performance of classifiers the accuracy metric is used because of using a balanced dataset and the accuracy is a good measure when data classes are nearly balanced. The accuracy metric is described as follows:

$$Accuracy = (TP + TN)/(TP + TN + FP + FN) \tag{2}$$

Where TP, TN, FP, and FN are true positive, true negative, false positive and false negative, respectively.The evaluation results are represented in table 2. As shown from table2, the best performance was achieved by the SVM classifier followed by the NB classifier followed by a logistic regression classifier.

TABLE 2: EVALUATION RESULTS

	Accuracy
NB	85.4%
SVM	86%
Logistic Regression	81.4%

5. Conclusion and Future Work

This paper provided new experiments to classify sentiment of large-scale data using Spark's MLlib by applying different classification algorithms, Naïve Bayes (NB), Support vector machine(SVM) and logistic regression. Apache spark machine library (MLlib) was used as it is scalable to handle a large volume of data. The experiments were made using Amazon product reviews dataset which contains four million reviews, 3600000 reviews for training and 40000 reviews for testing. To clean and prepare the data for classification, many preprocessing steps were applied. Three classifiers were compared in terms of accuracy, naïve Bayes, Support vector machine, and logistic regression. The experiments results showed that the support vector machine classifier has better performance than the other classifiers. As future work, further experiments will be conducted using different feature sets and n-gram models (bi-gram and tri-gram) that may enhance the performance of the classification.

References

[1] Banić, L., Mihanović, A., & Brakus, M. (2013, May). Using big data and sentiment analysis in product evaluation. In Information & Communication Technology Electronics & Microelectronics (MIPRO), 2013 36th International Convention on (pp. 1149-1154). IEEE.

[2] Benjamins, V. R. (2014, June). Big data: From hype to reality?. In *WIMS* (pp. 2-1).

[3] Sagiroglu, S., & Sinanc, D. (2013, May). Big data: A review. In Collaboration Technologies and Systems (CTS), 2013 International Conference on (pp. 42-47). IEEE.

[4] Chen, M., Mao, S., & Liu, Y. (2014). Big data: A survey. *Mobile networks and applications*, *19*(2), 171-209.

[5] Zheng, J., & Dagnino, A. (2014, October). An initial study of predictive machine learning analytics on large volumes of historical data for power system applications. In *Big Data (Big Data), 2014 IEEE International Conference on* (pp. 952-959). IEEE

[6] Zaharia, M., Chowdhury, M., Franklin, M. J., Shenker, S., & Stoica, I. (2010). Spark: Cluster computing with working sets. HotCloud, 10(10-10), 95.

[7] Zaharia, M., Xin, R. S., Wendell, P., Das, T., Armbrust, M., Dave, A. & Ghodsi, A. (2016). Apache spark: a unified engine for big data processing. Communications of the ACM, 59(11), 56-65.

[8] Meng, X., Bradley, J., Yavuz, B., Sparks, E., Venkataraman, S., Liu, D., ... & Xin, D. (2016). MLlib: Machine learning in apache spark. The Journal of Machine Learning Research, 17(1), 1235-1241.

[9] Medhat, W., Hassan, A., & Korashy, H. (2014). Sentiment analysis algorithms and applications: A survey. *Ain Shams Engineering Journal*, *5*(4), 1093-1113.

[10] Pang, B. and Lee, L. (2002). Thumbs up? Sentiment classification using machine learning techniques. Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing, 10:79–86.

[11] Nabil, M., Aly, M., & Atiya, A. (2015). Astd: Arabic sentiment tweets dataset. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing* (pp. 2515-2519).

[12] Duwairi, R. M., & Qarqaz, I. (2014, August). Arabic sentiment analysis using supervised classification. In *Future Internet of Things and Cloud (FiCloud), 2014 International Conference on*(pp. 579-583). IEEE.

[13] Bhatt, A., Patel, A., Chheda, H., & Gawande, K. (2015). Amazon Review Classification and Sentiment Analysis. *International Journal of Computer Science and Information Technologies*, *6*(6), 5107-5110.

[14] Kang, H., Yoo, S. J., & Han, D. (2012). Senti-lexicon and improved Naïve Bayes algorithms for sentiment analysis of restaurant reviews. *Expert Systems with Applications*, *39*(5), 6000-6010.

[15] Omar, N., Albared, M., Al-Shabi, A. Q., & Al-Moslmi, T. (2013). Ensemble of classification algorithms for subjectivity and sentiment analysis of Arabic customers' reviews. *International Journal of Advancements in Computing Technology*, *5*(14), 77.

[16] Sehgal, D., & Agarwal, A. K. (2016, November). Sentiment analysis of big data applications using Twitter Data with the help of HADOOP framework. In *System Modeling & Advancement in Research Trends (SMART), International Conference* (pp. 251-255). IEEE.

[17] Liu, B., Blasch, E., Chen, Y., Shen, D., & Chen, G. (2013, October). Scalable sentiment classification for big data analysis using naive bayes classifier. In *Big Data, 2013 IEEE International Conference on* (pp. 99-104). IEEE.

[18] Baltas, A., Kanavos, A., & Tsakalidis, A. K. (2016, August). An apache spark implementation for sentiment analysis on twitter data. In *International Workshop of Algorithmic Aspects of Cloud Computing* (pp. 15-25). Springer, Cham.

[19] Adib, P., Alirezazadeh, S., & Nezarat, A. (2017, October). Enhancing trust accuracy among online social network users utilizing data text mining techniques in apache spark. In *Computer and Knowledge Engineering (ICCKE), 2017 7th International Conference on* (pp. 283-288). IEEE.

[20] Pranckevičius, T., & Marcinkevičius, V. (2017). Comparison of naive bayes, random forest, decision tree, support vector machines, and logistic regression classifiers for text reviews classification. *Baltic Journal of Modern Computing*, *5*(2), 221.

[21] Zhang, X., Zhao, J., & LeCun, Y. (2015). Character-level convolutional networks for text classification. In *Advances in neural information processing systems* (pp. 649-657).

[22] Kotsiantis, S. B., Zaharakis, I., & Pintelas, P. (2007). Supervised machine learning: A review of classification techniques. *Emerging artificial intelligence applications in computer engineering*, *160*, 3-24.

[23] Kumar, A., & Sebastian, T. M. (2012). Sentiment analysis on twitter. *International Journal of Computer Science Issues (IJCSI)*, *9*(4), 372.