Proceedings of the 2nd International Conference on Computing Innovation and Applied Physics
DOI: 10.54254/2753-8818/18/20230402

Simulating real-time tweet sentiment analysis by different machine learning methods based on spark

Ertong Wei

University of Toronto, Toronto, Ontario, Canada, M5S 1A4

ertong.wei@mail.utoronto.ca

Abstract. Sentiment analysis is essential since it benefits many fields, such as politics and economics. Because much data is generated every moment, a real-time processing system can efficiently analyze sentiment. This paper uses Spark to simulate real-time tweet sentiment analysis, and compares the performances of three machine learning methods, Logistic Regression, Naive Bayes, and Decision Tree. The idea of the real-time tweet sentiment analysis system is using Spark Streaming to send a batch of tweets every fixed period to a machine learning pipeline to predict the emotions of tweets. In the pipeline, tweets will be tokenized first, then the stop words in tweets will be removed. After that, the author uses TF-IDF to extract features, transferring data from unstructured to structured. The last stage is using the machine learning method to predict the sentiments of tweets. By comparing, Logistic Regression has the best performance, and the second one is Naive Bayes, Decision Tree performs not as well as the other two methods.

Keywords: spark streaming, tweet, sentiment analysis, real-time, machine learning pipeline, logistic regression, naive bayes, decision tree

1. Introduction

The Internet has gradually become a significant part of people's lives in recent years. Amazon, Netflix, Twitter, and other social media have many users, and people communicate with each other online or express their ideas. Analyzing users' sentiments is necessary because it can be applied in many fields. For instance, public opinion analysis can help the government to learn people's tendencies about real-time hot events, and sentiment analysis can also be used in online shopping reviews to extract customers' feelings about products. Besides, the election result can even be predicted from political posts [1]. Artificial intelligence is a convenient choice for sentiment analysis because it consumes a lot of time and manpower if judging each piece of information by human beings. Sentiment analysis is an imperative branch of NLP (natural language processing), and there are many approaches to measure sentiment. The most original one is to label a group of words with positive or negative, then determine the sentiment by the prevalence of labeled words [2]. Tweet sentiment analysis is the focus of this paper. There are approximately 6000 tweets per sec, which is enormous in terms of volume and velocity for traditional data processing systems to handle [3]. Spark is a suitable platform for processing big data. There are two main reasons, fast speed and robust versatility. First, Spark is 100x faster than Hadoop when processing large-scale data in memory. Additionally, Spark can do real-time stream processing (Spark Streaming), machine learning (Spark MLlib), graph computation (Spark

DOI: 10.54254/2753-8818/18/20230402

GraphX), etc. In this paper, Spark Streaming and MLlib will be used. Spark Streaming is a stream processing framework on Spark, which can realize high-throughput, high-fault-tolerant real-time computing for massive data, and Spark MLlib can implement some standard machine learning algorithms and utilities. This research will simulate real-time stream tweets by Spark Streaming, and then compare the performance of different machine learning methods on tweet sentiment analysis.

2. Literature review

Many previous researchers have gained outstanding achievements in sentiment analysis. Samar et al. have compared the performance of SVM (Support Vector Machine), NB (Naive Bayes), and LR (Logistic Regression) on sentiment analysis for online reviews under Apache Spark [4]. They first remove all invalid data, then apply tokenization, then remove irrelevant parts and stop words, then convert text to vector by TF-IDF (term frequency-inverse document frequency), and SVM gets a better accuracy (86%) compared with NB (85.4%) and Logistic Regression (81.4%) [4]. Furthermore, another research on the Spark platform using KNN (K-Nearest Neighbors) to analyze sentiment and enrich the performance by using the Bloom filter to compress the storage size [5]. A sentiment analysis method based on BiLSTM (bidirectional long short term memory) performs better compared with RNN (Recurrent Neural Network), CNN (Convolutional Neural Network), LSTM (Long-Short Term Memory), and NB [6]. There are also some papers focusing on real-time sentiment analysis. Kilinc demonstrates that a considerable challenge of real-time sentiment analysis is the uncertainty of the reliability since there exist some fake accounts due to unethical reasons, he builds a spark-based real-time sentiment prediction framework that detects the authenticity of accounts before inputting data [7]. Moreover, SVM shows an accuracy of around 85% on real-time sentiment analysis of Amazon product reviews, there are four main steps in processing data, which are tokenization, removing stop words, POS tagging, and stemming [8]. In addition, Elzayady et al. process a real-time sentiment analysis for Saudi by lexicon-based algorithm [9].

3. Methodology

3.1. Data

3.1.1. Data analysis. The data used in this research comes from Kaggle [10]. The dataset is an entity-level tweet sentiment analysis dataset of Twitter that contains 74,681 data in total, each of which contains four parameters: id, entity, sentiment, and tweet. There are four classes of sentiment, positive, negative, neutral, and irrelevant, representing that the tweet is unconnected with the entity. Moreover, based on the limitation of Twitter, each tweet contains no more than 280 characters.

3.1.2. Data cleaning. After dropping data where the tweet is null, the amount of data is 73,995. Since this paper only focuses on sentiment analysis, the author drops the id, entity column, and the data where sentiment is irrelevant. Then there are 61,120 data left, and each row of data consists of sentiment (positive, negative, neutral) and tweet. It is unnecessary to over-sample or under-sample the data since this is a balanced dataset based on sentiment details, as shown in Table 1. Then randomly splitting the data to 70% training data and 30% testing data, and the distributions of training (positive: 14,349; negative: 12,727; neutral: 15,645) and testing data (positive: 6,306; negative: 5,381; neutral: 6,713) are also balanced.

DOI: 10.54254/2753-8818/18/20230402

Table 1. Sentiment distribution of the whole dataset(credit: original).

Sentiment	Amount
Positive	20654
Negative	22358
Neutral	18108

3.2. Model architecture

There are two core parts in this system: Spark Streaming and the machine learning pipeline. Spark Streaming helps to simulate real-time tweet data streaming. The machine learning pipeline is a workflow with a specific order and contains many sequential stages (transformers and predictors) from inputting data to the training model and predicting the output. The pipeline mechanism realizes the streaming encapsulation and management of all steps. The pipeline is constructed in advance, using Spark Streaming to input a batch of tweets for each fixed time to the machine learning pipeline to simulate real-time data. Once the pipeline receives the data, it processes the tweet to analyze the sentiment and sends it back to Spark Streaming.

3.2.1. Machine learning pipeline. The machine learning pipeline consisted of four main stages, tokenization, removing stop words, feature vectorization, and constructing the machine learning model.

3.2.1.1. Tokenization. Tokenization (word segmentation) is the way that split a piece of text into small tokens. The processing of tokenization is demonstrated in Figure 1.

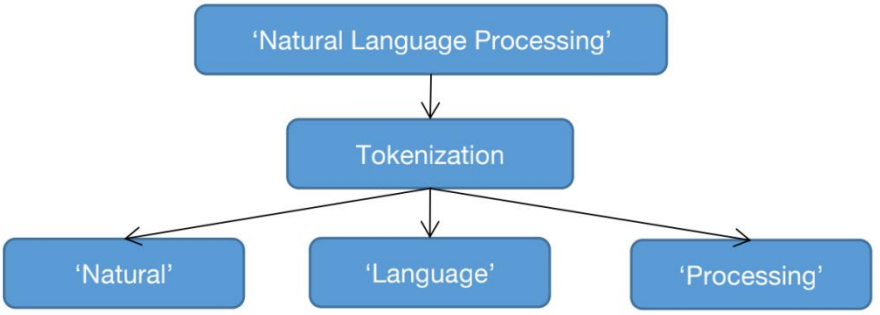

Figure 1. The processing of tokenization(credit: original).

3.2.1.2. Removing stop words. While judging the sentiment of tweets, disposing of stop words can make sentiment analysis focus on meaningful words. Spark provides the StopWordsRemover function, which can directly remove all stop words. For example, given the sentence 'i like nlp and ml', the result will be 'like nlp ml' after applying StopWordsRemover. 'i' and 'and' have been removed as stop words, and the two words indeed have nothing to do with predicting the sentiment of the text 'i like nlp and ml'.

3.2.1.3. Feature vectorization. Text information is unstructured, and most machine learning methods input structured data. Thus, to enable the machine to learn efficiently, tweets need to be converted to vectors before training the machine learning model. The feature extraction method used in this paper is TF-IDF. TF is short for term frequency-inverse document frequency. It is used to measure the

DOI: 10.54254/2753-8818/18/20230402

importance of a word to the text. The significance of a word is proportional to its frequency in the current text and inversely proportional to its frequency in other texts in the corpus. The formula of TF and IDF is shown below, then TF-IDF = TF * IDF.

$$\text{TF} = \frac{\text{the amount of times a word appears in the text}}{\text{total number of words in the text}} \quad (1)$$

$$\text{IDF} = \log \frac{\text{the number of all documents in the corpus}}{\text{the number of documents contain the word + 1}} \quad (2)$$

3.2.1.4. Classifier. The data has already been transformed into the structured data in the previous feature extraction stage, connecting the feature extraction output with the machine learning model. Three machine learning models from Spark MLlib are used separately in the pipeline: Logistic Regression, Naive Bayes, and Decision Tree. The details of the three methods will be described in the after part. Then, use training data to train the pipeline in advance.

3.2.2. Spark streaming. Although Spark Streaming supports real-time processing of streaming data and can collect data in real time, this experiment uses it to simulate real-time data since the data used is not real-time. The data type of the testing data is data frame after reading from the CSV file. Firstly, it is necessary to convert the testing data from data frame type to RDD (Resilient Distributed Dataset), which is the most basic abstraction in Spark, RDD is an immutable, partitionable collection, and the elements in RDD can be computed in parallel. After that, Spark Streaming split the data into n small batches. Then process each small batch of data by Spark Streaming in a specific time interval to simulate real-time data. In this research, the author chooses n = 10 and lets Spark Streaming send DStream (a batch of RDDs in a time interval) every second to the machine learning pipeline to analyze tweet sentiment, predicting the sentiment of roughly between 1,000 and 1,300 tweets per second until all RDD has been sent to the pipeline.

3.3. Machine learning methods

3.3.1. Logistic regression. Logistic regression is mainly used for classification problems, especially binary classification problems (0/1, yes/no, true/false), by applying the sigmoid function on linear regression, mapping the unbounded output range of the linear regression to between 0 and 1 to predict the probability of an event.

3.3.2. Naive bayes. The Naive Bayes algorithm is based on Bayes' theorem, the formula of Bayes' theorem as displayed below.

$$\text{P(class | features)} = \frac{\text{P(features | class) P(class)}}{\text{p(features)}} \quad (3)$$

In tweet sentiment analysis, the class represents positive/negative/neutral, and the features contain more than one feature. The classification process uses the given features to calculate the probability of P(class|features). For example, if there are only two sentiments, positive and negative, and P(positive|features) = 0.667, P(negative|features) = 0.333, then the result predicted by the Naive Bays classifier is positive.

3.3.3. Decision tree. The Decision Tree is a tree structure model. In a decision tree, each internal node is a judgment on a condition, each branch represents the output of the condition of the previous node, and each leaf node represents the final output of the decision tree. In a binary tweet sentiment analysis, the result of a leaf node is positive or negative; in ternary sentiment analysis, there is one more possible output --- neutral. The maximum depth of the decision tree in this research is 30.

3.4. Results

First, compare the performance of different machine learning methods in tweet binary emotion

DOI: 10.54254/2753-8818/18/20230402

analysis. Only tweets with positive or negative emotions are used to train the machine learning model. In addition, about 1300 tweets are inputted into the pipeline every second to analyze the sentiment. The accuracy of different machine learning approaches is shown in Table 2. According to the result, it is evident that Logistic Regression shows a better performance, then Naive Bayes is followed by, Decision Tree performs not as well as the other two machine learning methods.

Table 2. Binary sentiment analysis results(credit: original).

Machine learning method	Accuracy
Logistic Regression	88.689%
Naive Bayes	85.590%
Decision Tree	75.929%

After that, the author contrasts the performance on ternary sentiment analysis and inputting tweets with the positive, negative, or neutral sentiment to the machine learning pipeline. The Spark Streaming sends testing data to the pipeline about 1800 tweets each second. Table 3 demonstrates the accuracy of different methods. It is noticeable that all the performances of the three models experience a decrease after adding one more type of sentiment, and all of their accuracies decreased by around 10%. Besides, Logistic Regression still performs better than the Naive Bayes and Decision Tree.

Table 3. Ternary sentiment analysis results(credit: original).

Machine learning method	Accuracy
Logistic Regression	81.881%
Naive Bayes	75.485%
Decision Tree	62.517%

4. Conclusion

This research uses Spark Streaming and the machine learning pipeline to simulate real-time tweet sentiment analysis and compare the performance of different machine learning methods. In the tweet binary sentiment analysis, where tweets only have two possible sentiments, positive and negative, Spark Streaming sends around 1,300 tweets per second to the machine learning pipeline. Logistic Regression performs the best with an accuracy of 88.689%, and Naive Bayes performs better than the Decision Tree. Their accuracies are 85.590% and 75.929%, respectively. Under the situation that there are three possible sentiments for each tweet, the machine learning pipeline receives approximately 1,800 tweets every second, and the accuracy of Logistic Regression, Naive Bayes, and Decision Tree are 81.881%, 75.485%, 62.517% separately. All three models experience a fall in performance with one more emotion on real-time tweet sentiment analysis. In the future, the author will find real-time data to analyze the truly real-time tweet sentiment instead of simulating one and build a more accurate model for analyzing tweet sentiment. Moreover, future research will also focus on inputting more data for each second since, in the real world, there are roughly 6,000 tweets generated every second, but in the current research, only inputting less than 2,000 tweets per second due to the lack of data.

Acknowledgment

First of all, I am grateful to the professors at the university for teaching me professional knowledge, which played a significant role in completing this research. Secondly, I would like to thank my parents and friends for their support and encouragement and for helping me in life.

References

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

[2] Stine, R. A. (2019). Sentiment analysis. Annual review of statistics and its application, 6,

DOI: 10.54254/2753-8818/18/20230402

287-308.

[3] Nair, L. R., Shetty, S. D., & Shetty, S. D. (2017). Streaming big data analysis for real-time sentiment based targeted advertising. International Journal of Electrical and Computer Engineering, 7(1), 402.

[4] Al-Saqqa, S., Al-Naymat, G., & Awajan, A. (2018). A large-scale sentiment data classification for online reviews under apache spark. Procedia Computer Science, 141, 183-189.

[5] Nodarakis, N., Sioutas, S., Tsakalidis, A. K., & Tzimas, G. (2016, March). Large Scale Sentiment Analysis on Twitter with Spark. In EDBT/ICDT Workshops (pp. 1-8).

[6] Xu, G., Meng, Y., Qiu, X., Yu, Z., & Wu, X. (2019). Sentiment analysis of comment texts based on BiLSTM. Ieee Access, 7, 51522-51532.

[7] Kılınç, D. (2019). A spark-based big data analysis framework for real-time sentiment prediction on streaming data. Software: Practice and Experience, 49(9), 1352-1364.

[8] Jabbar, J., Urooj, I., JunSheng, W., & Azeem, N. (2019, May). Real-time sentiment analysis on E-commerce application. In 2019 IEEE 16th International Conference on Networking, Sensing and Control (ICNSC) (pp. 391-396). IEEE.

[9] Assiri, A., Emam, A., & Al-Dossari, H. (2016, December). Real-time sentiment analysis of Saudi dialect tweets using SPARK. In 2016 IEEE International Conference on Big Data (Big Data) (pp. 3947-3950). IEEE.

[10] Twitter Sentiment Analysis. https://www.kaggle.com/ datasets/jp797498e/twitter- entity -sentiment-analysis.