# An Apache Spark Implementation for Sentiment Analysis on Twitter Data

Alexandros Baltas, Andreas Kanavos(✉), and Athanasios K. Tsakalidis

Computer Engineering and Informatics Department,
University of Patras, Patras, Greece
{ampaltas,kanavos,tsak}@ceid.upatras.gr

**Abstract.** Sentiment Analysis on Twitter Data is a challenging problem due to the nature, diversity and volume of the data. In this work, we implement a system on Apache Spark, an open-source framework for programming with Big Data. The sentiment analysis tool is based on Machine Learning methodologies alongside with Natural Language Processing techniques and utilizes Apache Spark's Machine learning library, MLlib. In order to address the nature of Big Data, we introduce some pre-processing steps for achieving better results in Sentiment Analysis. The classification algorithms are used for both binary and ternary classification, and we examine the effect of the dataset size as well as the features of the input on the quality of results. Finally, the proposed system was trained and validated with real data crawled by Twitter and in following results are compared with the ones from real users.

**Keywords:** Apache Spark · Big Data · Classification · Microblogging · Sentiment Analysis · Social media analytics

## 1 Introduction

Nowadays people share moments, experiences and feelings through social networks. Microblogging platforms, namely Twitter, have recently become very popular. Founded in 2006, Twitter is a service which allows users to share 140-character posts. Having gained massive popularity while being widely considered one of the most influential services on the World Wide Web. Twitter has resulted in hosting massive datasets of information. Thus its data is gaining increasing interest. People use Twitter to share experiences and emotions with their friends about movies, products, events etc., so a system that extracts sentiments through an online community may have many real-life applications such as recommendation systems. This enormously continuous stream of Twitter data posts reflects the users opinions and reactions to phenomena from political events all over the world to consumer products [20]. It is well pointed that Twitter posts relate to the user's behavior and often convey substantial information about their emotional state [3].

T. Sellis and K. Oikonomou (Eds.): ALGOCLOUD 2016, LNCS 10230, pp. 15–25, 2017.
DOI: 10.1007/978-3-319-57045-7_2

Unlike other networks, users' posts in Twitter have some special characteristics. The short length that the posts are allowed to have, results in more expressive emotional statements. Analyzing tweets and recognizing their emotional content is a very interesting and challenging topic in the microblogging area. Recently many studies have analyzed sentiment from documents or web-related content, but when such applications are focused on microblogging, many challenges occur. The limited size of the messages, along with the wide range of subjects discussed, make sentiment extraction a difficult process. Concretely, researchers have used long-known machine learning algorithms in order to analyze sentiments. So the problem of sentiment extraction is transformed into a classification problem. Datasets of classified tweets are used to train classifiers which in following are used to extract the sentiments of the messages.

In the meantime, as data grows, cloud computing evolves. Frameworks like Hadoop, Apache Spark, Apache Storm and distributed data storages like HDFS and HBase are becoming popular, as they are engineered in a way that makes the process of very large amounts of data almost effortless. Such systems evolve in many aspects, and as a result, libraries, like Spark's MLlib that make the use of Machine Learning techniques possible in the cloud, are introduced.

In this paper we aim on creating a Sentiment Analysis tool of Twitter data based on Apache Spark cloud framework, which classifies tweets using supervised learning techniques. We experiment with binary and ternary classification, and we focus on the change in accuracy caused by the training dataset size, as well as the features extracted from the input.

The remainder of the paper is structured as follows: Section 2 presents the related work. Section 3 presents cloud computing methodologies, while Sect. 4 presents the classification algorithms used in our proposed system. Section 5 presents the steps of training as well as the two types of classification, binary and ternary. Moreover, Sect. 6 presents the evaluation experiments conducted and the results gathered. Ultimately, Sect. 7 presents conclusions and draws directions for future work.

## 2 Related Work

In the last decade, there has been an increasing interest in studies of Sentiment Analysis as well as emotional models. This is mainly due to the recent growth of data available in the World Wide Web, especially of those that reflect people's opinions, experiences and feelings [17]. Sentiment Analysis is studied in many different levels. In [22], authors implement an unsupervised learning algorithm that classifies reviews, thus performing document level classification. In [13] authors operate in a word and sentence level, as they classify people's opinions. Moreover, Wilson et al. [24] operate on a phrase level, by determining the neutrality or polarity of phrases. Machine learning techniques are frequently used for this purpose. Pang et al. [18] used Naive Bayes, Maximum Entropy and SVM classifiers so as to analyze sentiment of movie reviews. Boiy and Moens [2] utilized classification models with the aim of mining the sentiment out of multilingual web texts.

Twitter data are used by researchers in many different areas of interest. In [8], Tweets referring to Hollywood movies are analyzed. They focused on classifying the Tweets and in following on analyzing the sentiment about the Hollywood movies in different parts of the world. Wang et al. [23] used a training data of 17000 Tweets in order to create a real-time Twitter Sentiment Analysis System of the U.S. 2012 Presidential Election Cycle. In addition, in [15], authors present a novel method for Sentiment Learning in the Spark framework; the proposed algorithm exploits the hashtags and emoticons inside a tweet, as sentiment labels, and proceeds to a classification procedure of diverse sentiment types in a parallel and distributed manner.

Other studies that investigate the role of emoticons on sentiment analysis of Tweets are the ones in [19,25]. In both works, Lexicons of Emoticons are used to enhance the quality of the results. Authors in [4] propose a system that uses an SVM classifier alongside a rule-based classifier so as to improve the accuracy of the system. In addition, in [3], authors utilized the Profile of Mood States psychometric method for analyzing Twitter posts and reached the conclusion that "the events in the social, political, cultural and economic sphere do have significant, immediate and highly specific effect on the various dimensions of public mood". Commercial companies and associations could exploit Twitter for marketing purposes, as it provides an effective medium for propagating recommendations through users with similar interests.

There is a lot of research interest in studying different types of information dissemination processes on large graphs and social networks. Naveed et al. [14] analyze tweet posts and forecast for a given post the likelihood of being retweeted on its content. Authors indicate that tweets containing negative emoticons are more likely to be retweeted than tweets with positive emoticons. Finally, previous works regarding emotional content are the ones in [9–12]; they presented various approaches for the automatic analysis of tweets and the recognition of the emotional content of each tweet based on Ekman emotion model, where the existence of one or more out of the six basic human emotions (Anger, Disgust, Fear, Joy, Sadness and Surprise) is specified.

## 3 Cloud Computing

### 3.1 MapReduce Model

MapReduce is a programming model which enables the process of large datasets on a cluster using a distributed and parallel algorithm [6]. A MapReduce program consists of 2 main procedures, Map() and Reduce() respectively, and is executed in 3 steps; Map, Shuffle and Reduce. In the Map phase, input data is partitioned and each partition is given as an input to a worker that executes the map function. Each worker processes the data and outputs key-value pairs. In the Shuffle phase, key-value pairs are grouped by key and each group is sent to the corresponding Reducer. Apache Hadoop is a popular open source implementation of the Map Reduce model.

### 3.2 Spark Framework

Spark Framework[1] is a newer framework built in the same principles as Hadoop. While Hadoop is ideal for large batch processes, it drops in performance in certain scenarios, as in iterative or graph based algorithms. Another problem of Hadoop is that it does not cache intermediate data for faster performance but instead, it flushes the data to the disk between each step. In contrast, Spark maintains the data in the workers' memory and as a result it outperforms Hadoop in algorithms that require many operations. Spark offers API in Scala, Java, Python and R and can operate on Hadoop or standalone while using HDFS, Cassandra or HBase.

### 3.3 MLlib

Spark's ability to perform well on iterative algorithms makes it ideal for implementing Machine Learning Techniques as, at their vast majority, Machine Learning algorithms are based on iterative jobs. MLlib[2] is Apache Spark's scalable machine learning library and is developed as part of the Apache Spark Project. MLlib contains implementations of many algorithms and utilities for common Machine Learning techniques such as Clustering, Classification, Regression.

## 4 Machine Learning Techniques

In this work, we utilized three classification algorithms in order to implement the Sentiment Analysis Tool. We examined both Binary and Ternary Classification on different datasets. On the Binary Classification case, we focus on the way that the dataset size affects the results, while on the Ternary Classification case, the focus is given on the impact of the different features of the feature vector given as an input to the classifier. The three algorithms utilized are Naive Bayes, Logistic Regression and Decision Trees.

### 4.1 Naive Bayes

Naive Bayes is a simple multiclass classification algorithm based on the application of Bayes' theorem. Each instance of the problem is represented as a feature vector, and it is assumed that the value of each feature is independent of the value of any other feature. One of the advantages of this algorithm is that it can be trained very efficiently as it needs only a single pass to the training data. Initially, the conditional probability distribution of each feature given class is computed, and then Bayes' theorem is applied to predict the class label of an instance.

[1] http://spark.apache.org/.

[2] http://spark.apache.org/mllib/.

### 4.2 Logistic Regression

Logistic regression is a regression model where the dependent variable can take one out of a fixed number of values. It utilizes a logistic function to measure the relationship between the instance class, and the features extracted from the input. Although widely used for binary classification, it can be extended to solve multiclass classification problems.

### 4.3 Decision Trees

The decision tree is a classification algorithm that is based on a tree structure whose leaves represent class labels while branches represent combinations of features that result in the aforementioned classes. Essentially, it executes a recursive binary partitioning of the feature space. Each step is selected greedily, aiming for the optimal choice for the given step by maximizing the information gain.

## 5 Implementation

The overall architecture of the proposed system is depicted in Fig. 1 taking into account the corresponding modules of our approach. Inititally, a pre-processing step, as shown in following subsection, is utilized and in following the classifiers for estimating the sentiment of each tweet, are used.

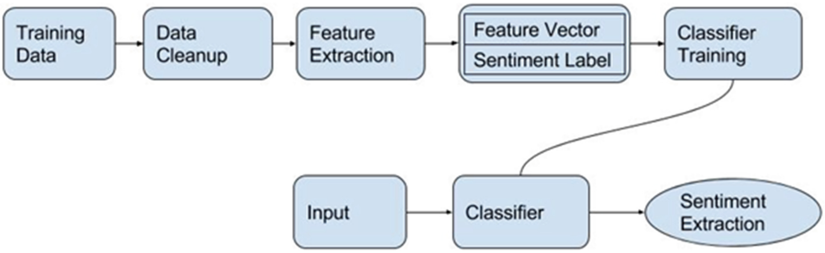

**Fig. 1.** Proposed system architecture

### 5.1 Binary Classification

For the Binary Classification, we used a dataset[3] of 1.578.627 pre-classified tweets as Positive or Negative. We split the original dataset into segments of 1.000, 2.000, 5.000, 10.000, 15.000, 20.000 and 25.000 tweets. Then for each segment, all metadata were discarded and each tweet was transformed to a vector of unigrams; unigrams are the frequencies of each word in the tweets.

[3] http://thinknook.com/twitter-sentiment-analysis-training-corpus-dataset-2012-09-22/.

### 5.2 Ternary Classification

Regarding Ternary Classification, we used two datasets[4] that were merged into one which eventually consists of 12.500 tweets. In the original datasets, each row contains the tweet itself, the sentiment, and other metadata related to the corresponding tweet. During the preprocessing, all irrelevant data were discarded, and we only used the actual text of the tweet, as well as the label that represents the sentiment; positive, negative or neutral.

Each tweet is then tokenized and processed. Occurrences of usernames and URLs are replaced by special tags and each tweet is finally represented as a vector which consists of the following features:

- **Unigrams**, which are frequencies of words occurring in the tweets.
- **Bigrams**, which are frequencies of sequences of 2 words occurring in the tweets.
- **Trigrams**, which are frequencies of sequences of 3 words occurring in the tweets.
- **Username**, which is a binary flag that represents the existence of a user mention in the tweet.
- **Hashtag**, which is a binary flag that represents the existence of a hashtag in the tweet.
- **URL**, which is a binary flag that represents the existence of a URL in the tweet.
- **POS Tags**, where we used the Stanford NLT MaxEnt Tagger [21] to tag the tokenized tweets and the following are counted:
    1. Number of Adjectives
    2. Number of Verbs
    3. Number of Nouns
    4. Number of Adverbs
    5. Number of Interjections

Then the ratios of the aforementioned numbers to the total number of tokens of each tweet are computed.

## 6 Evaluation

The results of our work are presented in the following Tables 1, 2, 3, 4 and 5. F-Measure is used as the evaluation metric of the different algorithms. For the binary classification problem (Table 1), we observe that Naive Bayes performs better than Logistic Regression and Decision Trees. It is also obvious that the dataset size plays a rather significant role for Naive Bayes, as the F-Measure value rises from 0.572 for a dataset of 1.000 tweets to 0.725 for the dataset of 25.000 tweets. On the contrary, the performance of Logistic Regression and Desicion Trees is not heavily affected by the amount of the tweets in the dataset.

[4] https://www.crowdflower.com/data-for-everyone/.

**Table 1.** Binary Classification - F-Measure

| Dataset size | Naive Bayes | Logistic Regression | Decision Trees |
|---|---|---|---|
| 1000 | 0.572 | 0.662 | 0.597 |
| 5000 | 0.684 | 0.665 | 0.556 |
| 10000 | 0.7 | 0.649 | 0.568 |
| 15000 | 0.71 | 0.665 | 0.575 |
| 20000 | 0.728 | 0.651 | 0.59 |
| 25000 | 0.725 | 0.655 | 0.56 |

**Table 2.** Ternary Classification - F-Measure

| Classifier | Positive | Negative | Neutral | Total |
|---|---|---|---|---|
| Naive Bayes | 0.717 | 0.75 | 0.617 | 0.696 |
| Logistic Regression | 0.628 | 0.592 | 0.542 | 0.591 |
| Decision Trees | 0.646 | 0.727 | 0.557 | 0.643 |

**Table 3.** Ternary Classification - F-Measure for Naive Bayes

| Features | Positive | Negative | Neutral | Total |
|---|---|---|---|---|
| Complete feature vector | 0.717 | 0.75 | 0.617 | 0.696 |
| w/o Unigrams | 0.628 | 0.602 | 0.537 | 0.592 |
| w/o Bigrams | 0.714 | 0.769 | 0.629 | 0.705 |
| w/o Trigrams | 0.732 | 0.77 | 0.643 | 0.716 |
| w/o User | 0.718 | 0.751 | 0.618 | 0.698 |
| w/o Hashtag | 0.721 | 0.739 | 0.608 | 0.692 |
| w/o URL | 0.72 | 0.748 | 0.619 | 0.697 |
| w/o POS Tags | 0.716 | 0.748 | 0.617 | 0.695 |

Regarding ternary classification, Naive Bayes outperforms the other two algorithms as well, as it can be seen in Table 2, with Linear Regression following in the results. Interestingly, unigrams seem to be the feature that boosts the classification performance more than all the other features we examine, while the highest performance is observed for the vectors excluding trigrams. Moreover, the binary field representing the existence of a hashtag in the tweet affects the results, as in all the experiments, the performance records smaller values without it. It can also be observed that all three algorithms perform better for positive and negative tweets than they do for neutral messages.

To further evaluate our system, we conducted a user study in which results from our approach were compared to those from user. The online survey using

**Table 4.** Ternary Classification - F-Measure for Logistic Regression

| Features | Positive | Negative | Neutral | Total |
|---|---|---|---|---|
| Complete feature vector | 0.628 | 0.592 | 0.542 | 0.591 |
| w/o Unigrams | 0.596 | 0.457 | 0.451 | 0.51 |
| w/o Bigrams | 0.616 | 0.6 | 0.546 | 0.59 |
| w/o Trigrams | 0.649 | 0.623 | 0.572 | 0.618 |
| w/o User | 0.625 | 0.6 | 0.54 | 0.592 |
| w/o Hashtag | 0.612 | 0.591 | 0.526 | 0.58 |
| w/o URL | 0.613 | 0.598 | 0.537 | 0.585 |
| w/o POS Tags | 0.646 | 0.585 | 0.512 | 0.587 |

**Table 5.** Ternary Classification - F-Measure for Decision Trees

| Features | Positive | Negative | Neutral | Total |
|---|---|---|---|---|
| Complete feature vector | 0.646 | 0.727 | 0.557 | 0.643 |
| w/o Unigrams | 0.57 | 0.681 | 0.549 | 0.597 |
| w/o Bigrams | 0.647 | 0.729 | 0.557 | 0.644 |
| w/o Trigrams | 0.646 | 0.728 | 0.557 | 0.644 |
| w/o User | 0.646 | 0.727 | 0.557 | 0.643 |
| w/o Hashtag | 0.639 | 0.601 | 0.529 | 0.594 |
| w/o URL | 0.64 | 0.615 | 0.554 | 0.606 |
| w/o POS Tags | 0.659 | 0.729 | 0.56 | 0.65 |

Ruby on Rails[5] contained 220 tweets of the test set of the dataset used for the ternary classification. 10 students associated with the University of Patras manually classified the tweets, and in following we compared the classification results of the best classifier to the users' responses. As the corresponding classifier, we choose Naive Bayes without Trigrams as it achieves the best F-Measure for all sentiments in ternary classification. Moreover, for each tweet, we selected the sentiment that appears the most in students' selections.

The percentages of corrected classified tweets are presented in following Fig. 2. We can observe that our proposed algorithm seems to achieve notable accuracy when dealing with neutral tweets, whereas positive and sentiment tweets do not have accurate precision. One possible explanation is the fact that the majority of the specific dataset contains tweets that are classified as neutral.

[5] http://sentipoll.herokuapp.com/.

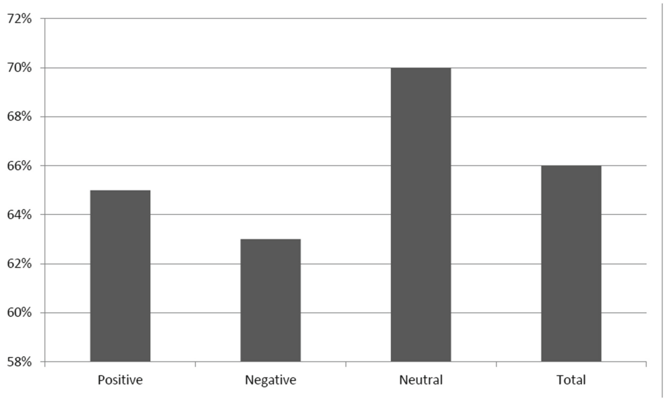

**Fig. 2.** Percentages of corrected classified tweets from Naive Bayes

## 7 Conclusions and Future Work

In our work, we have presented a tool that analyzes microblogging messages regarding their sentiment using machine learning techniques. More specifically, two datasets are utilized; a big dataset of tweets classified as positive or negative (binary), and a smaller one that consists of tweets classified as positive, negative or neutral (ternary). On the binary case, we examine the influence of the size of the dataset in relation with the performance of the sentiment analysis algorithms, while on the ternary case, we measure the system's accuracy regarding the different features extracted from the input. All the classification algorithms are implemented in Apache Spark cloud framework using the Apache Spark's Machine Learning library, entitled MLlib. Moreover, a user study was also conducted where University students manually classified tweets with the aim of validating our proposed tool accuracy.

As future work, we plan to further investigate the effect of different features on the input vector as well as utilize bigger datasets. Furthermore, we aim at experimenting with different clusters and evaluate Spark's performance in regards to time and scalability. Moreover, we plan on creating an online service that takes advantage of Spark Streaming, which is an Apache Spark's library for manipulating streams of data that provides users with real time analytics about sentiments of requested topics. Ultimately, personalization methods may be used to enhance the system's performance.

## References

1. Agarwal, A., et al.: Sentiment analysis of Twitter data. In: Workshop on Languages in Social Media (2011)
2. Boiy, E., Moens, M.-F.: A machine learning approach to sentiment analysis in multilingual web texts. Inf. Retrieval **12**(5), 526–558 (2008)

3. Bollen, J., Mao, H., Pepe, A.: Twitter sentiment and socio-economic phenomena. In: International Conference on Web and Social Media (ICWSM) (2011)
4. Chikersal, P., Poria, S., Cambria, E.: Sentiment analysis of tweets by combining a rule-based classifier with supervised learning. In: International Workshop on Semantic Evaluation (SemEval), pp. 647–651 (2015)
5. Chinthala, S., et al.: Sentiment analysis on twitter streaming data. in: Emerging ICT for Bridging the Future-Proceedings of the Annual Convention of the Computer Society of India (CSI), vol. 1 (2015)
6. Dean, J., Ghemawat, S.: MapReduce: simplified data processing on large clusters. Commun. ACM **51**(1), 107–113 (2008)
7. Go, A., Bhayani, R., Huang, L.: Twitter sentiment classification using distant supervision. CS224N Project Report, Stanford 1, vol. 12 (2009)
8. Hodeghatta, U.R.: Sentiment analysis of hollywood movies on Twitter. In: IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), pp. 1401–1404 (2013)
9. Kanavos, A., Perikos, I., Vikatos, P., Hatzilygeroudis, I., Makris, C., Tsakalidis, A.: Modeling ReTweet diffusion using emotional content. In: Artificial Intelligence Applications and Innovations (AIAI), pp. 101–110 (2014)
10. Kanavos, A., Perikos, I., Vikatos, P., Hatzilygeroudis, I., Makris, C., Tsakalidis, A.: Conversation emotional modeling in social networks. In: IEEE International Conference on Tools with Artificial Intelligence (ICTAI), pp. 478–484 (2014)
11. Kanavos, A., Perikos, I.: Towards detecting emotional communities in Twitter. In: IEEE International Conference on Research Challenges in Information Science (RCIS), pp. 524–525 (2015)
12. Kanavos, A., Perikos, I., Hatzilygeroudis, I., Tsakalidis, A.: Integrating user's emotional behavior for community detection in social networks. In: International Conference on Web Information Systems and Technologies (WEBIST) (2016)
13. Kim, S.M., Hovy, E.: Determining the sentiment of opinions. In: International Conference on Computational Linguistics, p. 1367 (2004)
14. Naveed, N., Gottron, T., Kunegis, J., Alhadi, A.C.: Bad news travel fast: a content-based analysis of interestingness on Twitter. Web Science, Article No. 8 (2011)
15. Nodarakis, N., Sioutas, S., Tsakalidis, A., Tzimas, G.: Large scale sentiment analysis on Twitter with Spark. In: EDBT/ICDT Workshops (2016)
16. Pak, A., Paroubek, P.: Twitter as a corpus for sentiment analysis and opinion mining. In: LREC, vol. 10 (2010)
17. Pang, B., Lee, L.: Opinion mining and sentiment analysis. Found. Trends Inf. Retrieval **2**(1–2), 1–135 (2008)
18. Pang, B., Lee, L., Vaithyanathan, S.: Thumbs up? Sentiment classification using machine learning techniques. In: ACL Conference on Empirical methods in Natural Language Processing, pp. 79–86 (2002)
19. Poonam, W.: Twitter sentiment analysis with emoticons. Int. J. Eng. Comput. Sci. **4**(4), 11315–11321 (2015)
20. Suttles, J., Ide, N.: Distant supervision for emotion classification with discrete binary values. In: CICLing, pp. 121–136 (2013)
21. Toutanova, K., Klein, D., Manning, C., Singer, Y.: Feature-rich part-of-speech tagging with a cyclic dependency network. In: HLT-NAACL, pp. 252–259 (2003)
22. Turney, P.D.: Semantic orientation applied to unsupervised classification of reviews. In: Annual Meeting on Association for Computational Linguistics, pp. 417–424 (2002)

23. Wang, H., Can, D., Kazemzadeh, A., Bar, F., Narayanan, S.: A system for real-time twitter sentiment analysis of 2012 us presidential election cycle. In: ACL System Demonstrations, pp. 115–120 (2012)
24. Wilson, T., Wiebe, J., Hoffmann, P.: Recognizing contextual polarity in phrase-level sentiment analysis. In: Conference on Human Language Technology and Empirical Methods in Natural Language Processing, pp. 347–354 (2005)
25. Yamamoto, Y., Kumamoto, T., Nadamoto, A.: Role of emoticons for multidimensional sentiment analysis of Twitter. In: International Conference on Information Integration and Web-based Applications Services (iiWAS), pp. 107–115 (2014)

http://www.springer.com/978-3-319-57044-0

Algorithmic Aspects of Cloud Computing
Second International Workshop, ALGOCLOUD 2016,
Aarhus, Denmark, August 22, 2016, Revised Selected
Papers
Sellis, T.; Oikonomou, K. (Eds.)
2017, XII, 183 p. 41 illus., Softcover

ISBN: 978-3-319-57044-0