The Impact of Natural Language Preprocessing on Big Data Sentiment Analysis

Mariam Khader, Arafat Awajan, and Ghazi Al-Naymat
Computer Science, Princess Sumaya University for Technology, Jordan

Abstract: *The sentiment analysis determines peoples' opinions, sentiments and emotions by classifying their written text into positive or negative polarity. The sentiment analysis is important for many critical applications such as decision making and products evaluation. Social networks are one of the main sources of sentiment analysis. However, the huge volume of data produced by social networks requires efficient and scalable analysis techniques to be applied. The MapReduce proved its efficiency and scalability in handling big data, thus attracted many researchers to use the MapReduce as a processing framework. In this paper, a sentiment analysis method for big data is studied. The method uses the Naïve Bayes algorithm for classifying texts into positive and negative polarity. Several linguistic and Natural Language Processing (NLP)preprocessing techniques are applied on a Twitter data set, to study their impact on the accuracy of big data classification. The preformed experiments indicates that the accuracy of the sentiment analysis is enhanced by 5%, yielding an accuracy of 73% on the Stanford Sentiment data set.*

Keywords: *Big data, natural language processing, MapReduce framework, Naïve Bayes and sentiment analysis.*

Received September 27, 2018; accepted January 21, 2019

1. Introduction

Big data is a term for huge and complex datasets, which cannot be processed or analyzed by traditional systems. The main characteristics of big data are huge volume, complex variety and high velocity. As the volume and variety of big data are continuously increasing, new techniques are required to extract information from this data [22]. Using parallel frameworks such as Hadoop/MapReduce has attracted many researchers, because of its simplicity and scalability in handling large volume of datasets. In comparison with other parallel frameworks, MapReduce is the most used framework. Programmers prefer this framework, because they do not have to worry about networking issues or how the data should be distributed since all these configuration issues are handled by the MapReduce itself.

The sentiment analysis is used to extract valuable information from datasets. It determines the polarity of a given text based on the sentiments extracted from the text. As a result, the sentiment analysis defines the sentiments, opinions or attitudes of people from their written text [10]. It can be applied using machine learning techniques, lexicon based techniques or both [11].

The sentiment analysis has gained its popularity because of the huge amount of data that contains sentiments, especially the data produced by social networks [15]. Social networks are considered rich resources to study peoples' personality or opinion about a specific subject or product. Users can utilize sentiment analysis to find the best choice of services or products before purchasing them. Companies can also utilize it to analyze customers' opinion and satisfaction about their products [12]. For example, the Twitter social network contains millions of active users. Consequently, opinion and feeling posts on Twitter are producing huge amounts of data. These posts are studied by companies to help in improving customers' service and build quality products.

Social networks are rapidly growing in volume, variety and complexity. Analyzing such networks is expensive and time consuming task. In order to solve the issue of analyzing huge volume and variety of datasets, researchers started to utilize parallel analysis frameworks. Among these frameworks, MapReduce gained most attentions [7].

MapReduce is based on the MapReduce programming model and the Hadoop Distributed File System (HDFS). Using MapReduce, large amount of sentiment data can be processed in parallel, which achieves better performance and requires less processing time, in comparison with single-machine processing. Another reason for using MapReduce is its ability to handle variety types of data, which make it suitable for handling sentiment data.

The aim of this work is to study the impact of Natural Language Processing (NLP) and linguistic preprocessing on the accuracy of big data analysis. This work evaluates the impact of the preprocessing techniques using a case study of a sentiment analysis method based on the MapReduce framework. The method is evaluated using a Twitter dataset. The method employs the Naïve Bayes algorithm for classifying English tweets into positive and negative sentiments. Several linguistic and NLP preprocessing techniques are applied to observe their impact on the classification quality of the Naïve Bayes algorithm.

The applied NLP techniques are the tokenization, Part of Speech (PoS) Tagging and lemmatization. In addition, other linguistic preprocessing techniques are applied, such as removing stop words, removing Uniform Resource Locator (URL), removing other users' mention, removing numbers, and hashtags.

The sentiment analysis method is implemented using Apache Mahout, which is a framework for distributed collaborative filtering, classification and clustering algorithms. Apache Mahout is built based on Java on the top of Hadoop using the MapReduce model [18].

Using MapReduce as framework allows the method to be scalable by default. Since the classification of sentiments in a text does not depend on other texts, applying sentiments analysis in parallel frameworks can be achieved seamlessly without dependencies [8].

The used NLP and linguistic preprocessing techniques have enhanced the accuracy of the classification results. In general, the accuracy of the sentiment classification was increased by nearly 5%. The accuracy was slightly increased by removing stop words, hashtags, user's mention and numbers. The PoS tagging had the most effect on increasing the accuracy of the classification, while the lemmatization effects is much less.

The rest of this work is structured as follows: section 2 presents the literature review of sentiment analysis methods based on MapReduce, in section 3, the MapReduce architecture is explained, in section 4, experimental results are discussed and Finally, section 5 contains the conclusion remarks.

2. Literature Review

The importance of the sentiments analysis motivated many researchers to use techniques to enhance the efficiency of the analysis, especially using MapReduce to enhance processing scalability. In the literature, the sentiment analysis was applied using machine learning algorithms, lexicon based analysis or both [7].

Recently, several machine learning algorithms within Hadoop framework were utilized for sentiment analysis. Hadoop was used to enhance the analysis performance and the scalability in [1, 11, 19].

Researchers in [11] applied their own implementation of Naïve Bayes for classifying texts into positive and negative sentiments. Four modules of MapReduce code were developed, the Work Flow Controller (WFC), the data parser, the user terminal and the result collector. The method was applied on two data sets, the first was the Cornell University movie review; it contained 2000 positive and negative reviews. The second data set was the Amazon movie review data set; it contained five points rating system of positive and negative sentiments. The proposed approach has achieved high scalability and accuracy around 80%.

The authors of [1] have developed a sentiment analysis method based on Mahout for classifying texts into positive or negative sentiments.

Three techniques were used for text classification:

1. Naïve Bayes.
2. OpenNLP.
3. LMClassifier.

Although the performance of Naïve Bayes outperformed OpenNLP and LMClassifier, the method was applied using a combination of the three methods. The method was applied on two datasets; the first was the Sanders dataset and the Stanford Sentiment Test Set (STS-Test).

In [19], a sentiment analysis method based on Naïve Bayes was developed to classify texts into positive, negative or neutral. The method was applied on a movie dataset extracted from Twitter and utilized the SentiWordNet dictionary. That method enhanced the results of the classification by studying the effect of embedding the emoticons in the preprocessing step, by substituting each emoticons by a word that represents this emoticon. In addition, several preprocessing steps were applied including removal of additional white spaces, removal of special symbol, removal of URL's, removal of username, removal of hashtag and converting emoticons. Converting the emoticons in the preprocessing step has increased the accuracy results and reduced the number of neutral results.

In addition to Machine learning techniques, researchers have applied lexicon-based techniques for sentiment classification. Lexicon-based techniques rely on a sentiment lexicon for determining the sentiments. The lexicon-based techniques are categorized into be dictionary-based method or a corpus-based method, where the sentiment polarity is determined by statistical and semantic approaches [13].

Several techniques of lexicon-based analysis within Hadoop framework were proposed such as [5, 6, 14, 15, 20]. These techniques used different dictionaries such as Positive context, Negative context, Positive word, Negative word, Prohibited words, sentiment-bearing words, lexicon of specific events and AFINN dictionary.

The authors in [6], proposed a sentiment analysis method for classifying texts into positive, negative or natural sentiments. The method used four functions for performing the sentiment extraction. The functions were the polarity preprocessing, the Syntactic word analysis, the Morpheme analysis and the Prohibited words analysis. In addition, five dictionaries were used: Positive context, Negative context, Positive word, Negative word and Prohibited word. The method was applied on five datasets that were gathered from Twitter by the "Topsy" Application Programming

Interface (API). The accuracy results of the proposed method were close to manual analysis results.

The authors of [20] have proposed a dictionary based sentiment analysis method. The used dictionary consisted of sentiment-bearing words. The method was applied on a data set of tweets that was collected by the Flume application. The technique achieved good precision and accuracy results.

The sentiment analysis method in [5] was applied at the entity level for classifying texts into positive, negative or neutral. The Resource Description Framework (RDF) graphs was employed to classify a tweets lexicon for a sport event. The research focused on three analyses:

1. The relation between the number of tweets and the match scores during the championship.
2. The relation of the tweet polarity and the score of the match.
3. The relation of tweets number and the sentiment value.

Researchers in [15] developed a dictionary-based method for classifying sentiments into positive, negative or neutral sentiment. The researchers have utilized the AFINN dictionary by enriching it with synonymies using the WordNet. The system had four phases, the preprocessing, using of AFINN to determine words synonyms, calculating words weights and then classification. The polarity of the tweets was defined by calculating the sum of weights of all words in the tweet. The method achieved good results of classification rate and error rate.

The proposed sentiment analysis method in [14] classified texts using two approaches into positive/negative or positive/negative/neutral. The method applied several preprocessing techniques:

1. Removing stop words.
2. Removing numbers.
3. Removing URL and tokenization.

In the first approach, the classification was performed by calculating the similarity of words in each tweet with three documents, which define the two classes. The second approach was based on a new developed formula for calculating the semantic similarity between each word in the tweet and the two words: positive and negative. The method was applied on two datasets that were collected using Twitter4j and Flume. The algorithm has achieved high rates of accuracy, recall, precision and F1-score.

Besides machine learning and lexicon based techniques, hybrid technique was applied for sentiment analysis [8]. The authors in [8] proposed a system combined of lexicon builder and the Logistic Regression algorithm for sentiment classification. The used lexicon, was built using HBase framework, and contained opinion lexicon, which contained a list of positive and negative words for twitter. The system has achieved good rates of scalability and accuracy.

The previous mentioned sentiment analysis methods focused on utilizing the MapReduce parallel framework to increase the scalability of the sentiment analysis techniques. Most of the methods used datasets extracted from Twitter social network. The high accuracy of Naïve Bayes in classifying sentiments, made it the most used machine learning algorithm. However, the fast execution time of lexicon based techniques attracted many other researchers. NLP methods, such as semantic analysis, were used to increase the accuracy of the sentiment analysis.

In addition, preprocessing techniques, such as removing stop words, URL, stop words and numbers were used to enhance the sentiment classification task.

3. Hadoop/MapReduce Architecture

The Hadoop is a parallel framework for handling large data sets. It can be built of thousands of distributed machines to perform distributed processing. The framework is built based on the Master-Slave architecture. The master node is called NameNode; it stores metadata about the stored files and jobs running within the cluster. The actual data is stored on the DataNodes. The files are stored as sequence of blocks within the HDFS [2, 22].

The MapReduce is a programming paradigm for processing huge data. Each MapReduce job is performed in two consecutive phases: the map and reduce phases. MapReduce jobs read the input data from the HDFS and write the output back to the HDFS. The steps of a MapReduce job work is illustrated by Figure 1.

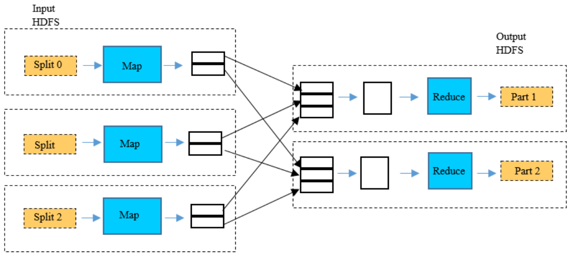

Figure 1. MapReduce job processing steps [1].

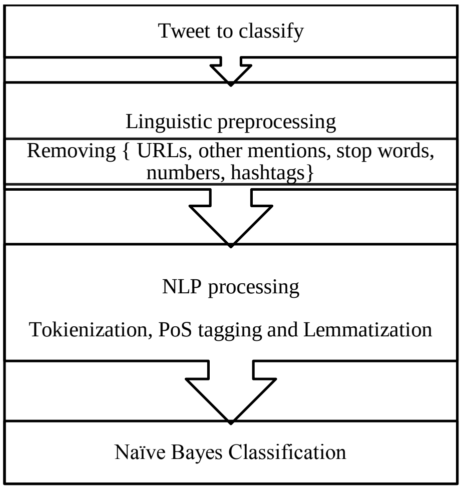
Tweet to classify
Linguistic preprocessing
Removing { URLs, other mentions, stop words, numbers, hashtags}
NLP processing
Tokienization, PoS tagging and Lemmatization
Naïve Bayes Classification

Figure 2. Steps of the proposed sentiment analysis method.

The input data is splitted into several parts to be processed independently by the Map nodes. The output of Map nodes is partitioned, shuffled/sorted and sent to reduce nodes.

4. Sentiment Analysis Method

This section presents the steps of the sentiment analysis method. The method classifies a twitter data set of English language into positive or negative sentiments.

The classification was mainly performed using Naïve Bayes algorithm, which is implemented in the Mahout framework.

In addition, several NLP and linguistic preprocessing techniques were applied on the tweets before classifying them. The use of NLP and linguistic preprocessing aims to enhance the accuracy of the Naïve Bayes in determining the sentiments. The steps of the sentiment analysis method is illustrated in Figure 2 and explained in the next subsections.

5. Linguistic Pre-processing

The linguistic preprocessing reduces the noise in the tweets and facilitates the classification process [3]. In the literature, different preprocessing types were applied such as negation handling, removing mentions, removing URLs, removing numbers, removing stop words, acronyms expansion and replacing emoticons with words [3, 14].

In the used data set, all the tweets were preprocessed before the training and testing phases. The linguistic preprocessing types that were performed in this study were:

1. *Removing Stop Word*: this phase removes the stops words, which has no effect on the classification process. It includes removing prepositions, article like "a", "an" and "the". The stop words do not contain or affect sentiments, thus deleting them refines the tweets.
2. *Removing URLs*: the URLs do not express any sentiment and do not affect the classification, thus removing them also will reduce the noise from the tweets.
3. *Removing Numbers*: numbers in the tweets do not have an effect on sentiments analysis and removing them reduces the noise in the tweets and enhances the efficiency.
4. *Removing Other Users' Mention*: in this step, other user's mention was removed, which has no effect on the polarity of the tweets.
5. *Removing Hashtags*: the hashtags (#) also has no effect on the classification process.

6. NLP Processing

After the preprocessing, the tokenization, PoS tagging and lemmatization are applied on the tweets. The tokenization splits the tweets into tokens (terms) by removing white spaces, commas or other symbols. It is an important step because it is needed for extracting words' PoS tags and lemmas.

The lemmatization is used to reduce any inflectional forms or derivationally related forms of a word and convert it into its basic form. For example, the sentence "The boy's cars are different colours" will be converted into "the boy car be differ colour". Table 1shows an example of word lemmatization output.

Table 1. Example of lemmatization results.

Word possibilities	Word basic form
am, are, is	be
car, cars, car's, cars'	car

The lemmatization takes as an input a word and its PoS tags and returns the "lemma" of the word. Each word may have different lemmas, by given the PoS tag, the right lemma can be found. For example, the lemma of the word "*saw*" is "*saw*" if its PoS was noun (NN) and its lemma is "*see*" if its PoS was a paste tense verb (VBD).

The Tokenization, Lemmatization and PoS tagging in this paper are performed using Apache OpenNLP, which is machine learning library for NLP tasks in the MapReduce. OpenNLP support the common NLP tasks like chunking, sentence segmentation, tokenization, part-of-speech tagging, named entity extraction, parsing and co reference resolution. It also includes a maximum entropy and perceptron based machine learning.

In OpenNLP, two types of lemmatization exist, a statistical and dictionary based lemmatization. The used lemmatization in this paper is the dictionary-based lemmatizer, which uses a dictionary that contains all valid and possible combination of the word and its lemma. Each combination contains{word, PoS tag and its corresponding lemma}. The dictionary is available online in [17].

The used PoS tags in OpenNlpare based on the tags defined in the Penn Tree Bank. An example of these tags is shown in Table 2; a complete list of these tags is available in [21]. In order to improve the Naïve Bayes classification, a term weighting method is used, which takes into account the PoS tagging. The focus is placed on adjectives, which is determined using PoS tagging.

This is because of the fact that adjectives can have a strong indicator of the sentiment in the text[16]. Thus, the adjectives are given higher weight than other terms in the text during the classification process.Besides adjectives, other terms are experimented to study their effect on the accuracy results. These terms are verbs and nouns.

7. Naïve Bayes Classification

The Naïve Bayes algorithm is a simple and efficient machine learning algorithm for text classification [9].

Table 2. Used PoS tags in OpenNLP (Treebank).

Tag	Meaning of tag
NN	noun, singular or mass
DT	determiner
VB	verb, base form
VBD	verb, past tense
VBZ	verb, third person singular present
IN	preposition or subordinating conjunction
NNP	proper noun, singular
TO –	the word “to”
JJ	adjective

The Naïve Bayes is a probabilistic classifier that applies the Bayes' theorem with independence presumption between features. Assuming a set of n documents D = $\{d_1, d_2,\ldots,d_n\}$, which belong to one of the m classes C = $\{c_1, c_2, \ldots, c_m\}$. The set of words W = $\{w_1, w_2, \ldots,w_x\}$ is a group of unique words, each appears once at least in one of the documents D. The probability of classifying a document d in class c is computed by the following Bayes’ rule:

$$P(c|d) = \frac{P(c)P(d|c)}{P(d)} \quad (1)$$

8. Experimental Results

This section presents the environment setups, the used data set and the conducted experiments for studying the method performance. In order to evaluate the quality of the sentiment analysis method, several experiments were conducted.

The experiments goal was to evaluate the effects of the NLP and linguistic processing steps on the accuracy of sentiment classification. Thus, the method has been run before and after applying NLP and linguistic preprocessing.

The goal of the experiments in this study was the following:

- Increasing the accuracy of the classification using the linguistic and NLP preprocessing.
- Determine the best performance of the Naïve Bayes classifier using PoS weighting.
- The sentiment analysis method is parallelized using the MapReduce framework, which ensure scalability and efficiency in handling large size of data.

8.1. Environment Setup

A comprehensive performance analysis of the method has been executed on 12network-connected machines. One machine was configured as NameNode, another machine isconfigured as Secondary NameNode and the rest of machines areconfigured as DataNodes. The DataNodes have8 vCPU and 8 GB Ram, the NameNode and Secondary DataNode had 24 vCPU and 16 GB RAM. All machines have run Hadoop version 2.6.2.0 on the top of CentOS 7as an operating system. The sentiment analysis method is implemented using a Javac Compiler of version 1.8 and utilized the Mahout Library version 2.6.2.0.

9. Data Set

For the experimental study, the algorithm has been evaluated on a real-world data set from Twitter. The dataset is the Stanford Twitter Sentiment data set [4], it contains 1600000 tweets for training, divided into positive and negative sentiments. Each row contains six fields:

1. Polarity of the tweet (0 for negative and 4 for positive).
2. Id.
3. Date.
4. Flag.
5. User.
6. The text of the tweet.

For this research, the concern is only on the first and last fields. The test dataset contains 498 records labelled “positive”, “negative” and “neutral”.

However, only “positive” and “negative” records from the test dataset are selected, which are 359 records splitted into 177 negative tweets and the rest are positive tweets.

10. Results

Although, the running time has been increased, the integration of the NLP and linguistic preprocessing in the classification process achieved improvements in the classification accuracy. Using PoS tagging, adjectives are specified and assigned more weight than other terms in the tweets. Since the used Naïve Bayes algorithm uses the Term Frequency–Inverse Document Frequency (TF-IDF) for calculating terms weight, more weight for adjective is done by increasing the term frequency. The following tweet demonstrates the effect of PoS term weighting on the classification accuracy.

"My dentist appt today was actually quite enjoyable"

The tweet is labelled as "positive" polarity; however, the Naïve Bayes has classified the tweet as negative. This is because the words "dentist" and "quite" in the Naïve Bayes model is assigned as representative for negative category (class).By assigning more weight for the adjective "enjoyable", the Naïve Bayes had successfully classified the tweet into positive polarity.

In order to specify the best performance for the Naïve Bayes classifier using PoS weighting, different experiments are performed using different weights for the adjectives. The best weight for adjectives is determined by experiments as shown in Table 3. The results indicated that the best accuracy values are achieved by increasing the adjectives frequency by 4.

Table 3. The accuracy results after using different weight for the adjective.

Added weight to adjectives	2	3	4	5	6
Accuracy	70.2%	72.1%	73.5%	73.5%	73.5%

Table 4. Accuracy results after using the NLP and linguistic processing steps.

	Without linguistic preprocessing	Removing numbers, URL, hashtags, users' mentions	Removing stop word
Naïve Bayes	68.5%	71.0%	70.7%
PoS Tagging	71.0%	72.7%	73.5%

Table 4 shows the accuracy results of the sentiment classification with and without linguistic and NLP preprocessing. The stop words are experimented in separate runs because they have the highest impact on the accuracy results. This experiment is performed using the 1600000 tweets for training. The best results of accuracy is achieved after removing the URL, stop words, other users' mentions and hashtags combined with increasing the weight of adjective.

Table 5 shows the accuracy results of the sentiment classification with and without linguistic preprocessing and lemmatization. This experiment is performed using 5000 tweets in the training phase and number of positive and negative tweets was equal. The results indicates a slight enhancement in the accuracy after removing numbers, URLs, users' mentions and hashtags combined with lemmatization. However, the lemmatization adds a large overhead on the execution time.

Table 6 shows the accuracy results after increasing the weight of different terms in the dataset. Each term weight was increased by 4 and ran in a separate experiment. The used terms were adjectives, nouns and verbs (including verbs in its base form, past tense and third person singular present verbs). This experiment is performed using 5000 tweets in the training phase.

Before increasing the weight of terms, the Naïve Bayes achieved classification accuracy of 68.5%. Among the three terms, increasing the weight of nouns achieved the best accuracy results followed by adjectives and verbs respectively. The last column of Table 4 shows the results of increasing the weight of all terms together, which has achieved better accuracy results.

Table 5. Accuracy results before and after using lemmatization.

	Without linguistic preprocessing	Removing numbers, URL, hashtags, users' mentions
lemmatization	70.2%	70.8%

Table 6. Accuracy results after weighting different terms.

	Increasing Adjectives (JJ) weight	Increasing Verbs (VB, VBD, VBZ) weight	Increasing Nouns (NN) weight	Increasing weights of JJ, VB, VBD, VBZ, NN
Accuracy	69.7%	69.4%	71.0%	71.3%

11. Conclusions

The increase number of users and data uploaded to the social networks has produced huge volume and variety of data types. The huge volume of data requires analysis techniques that are capable to scale up efficiently as the size of data increases. In addition, traditional analysis techniques need to be amended to handle the variety types of the produced data.

In this study, the effects of using linguistic and NLP processing techniques for big data analys is were evaluated using a sentiment analysis method. The sentiment analysis employed Naïve Bayes algorithm for classifying tweets into positive and negative sentiments using the MapReduce. In order to increase the accuracy of the Naïve Bayes classification, the data was preprocessed using tokenization, PoS tagging and lemmatization.

In addition, several preprocessing techniques were applied; removing stop words, URL, other users' mention, numbers and hashtags. The proposed method increased the accuracy of Naïve Bayes classification by 5% yielding an accuracy of 73%.

Future work of this research will include:

- Evaluating the impact of other linguistic preprocessing approaches, such as negation handling.
- Evaluating the impact of other NLP approaches such as stemming.
- Applying the preprocessing techniques on other languages such as Arabic language.

References

[1] Chauhan V. and Shukla A., "Sentimental Analysis of Social Networks using MapReduce and Big Data Technologies," *International*

Journal of Computer Science and Network, vol. 6, no. 2, pp. 120-130, 2017.

[2] Dean J. and Ghemawat S., "MapReduce: Simplified Data Processing on Large Clusters," *in Proceedings of the 6th Symposium on Operating Systems Design and Implementation*, San Francisco, pp. 137-150, 2004.

[3] Etaiwi W. and Naymat G.," The Impact of applying Different Preprocessing Steps on Review Spam Detection," *in Proceedings of 8th International Conference on Emerging Ubiquitous Systems and Pervasive Networks*, Lund, pp. 273-279, 2017.

[4] Go A., Bhayani R., and Huang L., https://www.kaggle.com/kazanova/sentiment140, Last Visited 2009.

[5] González C., García-Nieto J., Navas-Delgado I., and Aldana-Monte J.,"A Fine Grain Sentiment Analysis with Semantics in Tweets," *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 3, no. 6, pp. 22-28, 2016.

[6] Ha I., Back B., and Ahn B., "MapReduce Functions to Analyze Sentiment Information from Social Big Data," *International Journal of Distributed Sensor Networks*, vol. 11, no. 6, pp. 1-11, 2015.

[7] Khader M., Awajan A., and Al-Naymat G., "Sentiment Analysis Based on MapReduce: A Survey," *in Proceedings of the 10th International Conference on Advances in Information Technology*, Bangkok, 2018.

[8] Khuc V., Shivade C., Ramnath R., and Ramanathan J., "Towards Building Large-Scale Distributed Systems For Twitter," *in Proceedings of the 27th Annual ACM Symposium on Applied Computing*, Trento, pp. 459-464, 2012.

[9] Lewis D., "Naïve (Bayes) At Forty: the Independence Assumption in Information Retrieval," *in Proceedings of European Conference on Machine Learning*, Chemnitz, pp. 4-15, 1998.

[10] Liu B., *Sentiment Analysis: Mining Opinions, Sentiments, and Emotions*, Cambridge University Press, 2015.

[11] Liu B., Blasch E., Chen Y., Shen D., and Chen G., "Scalable Sentiment Classification for Big Data Analysis Using Naïve Bayes Classifier," *in Proceedings of IEEE International Conference on Big Data*, Silicon Valley, pp. 99-104, 2013.

[12] Liu B., Pozzi F., Fersini E., and Messina E., *Sentiment Analysis in Social Networks*, Elsevier Science and Technology, 2016.

[13] Madani Y., Bengourram J., Erritali M., Hssina B., and Birjali M., "Adaptive E-Learning using Genetic Algorithm and Sentiments Analysis in a Big Data System," *International Journal of Advanced Computer Science and Applications*, vol. 8, no. 8, pp. 394-403, 2017.

[14] Madani Y., Erritali M., and Bengourram J., "Sentiment Analysis using Semantic Similarity and Hadoop MapReduce," *Knowledge and Information Systems*, vol. 59, no. 2. pp. 413-436, 2019.

[15] Madani Y., Mohammed E., and Jamaa B., "A Parallel Semantic Sentiment Analysis," *in Proceedings of 3rd International Conference of Cloud Computing Technologies and Applications*, Rabat, pp. 1-6, 2017.

[16] Nicholls C. and Song, F., "Improving Sentiment Analysis With Part-Of-Speech Weighting," *in Proceedings of the 8thInternational Conference on Machine Learning and Cybernetics*, Hebei, pp. 1592-1597, 2009.

[17] Opennlp A., https://raw.githubusercontent.com/richardwilly98/elasticsearch-opennlp-auto-tagging/master/src/main/resources/models/en-lemmatizer.dict, Last Visited 2018.

[18] Owen S., Anil R., Dunning T., and Friedman E., *Mahout in Action*, Manning, 2011.

[19] Parveen H. and Pandey S., "Sentiment Analysis on Twitter Data-set using Naive Bayes Algorithm," *in Proceedings of 2nd International Conference on Applied and Theoretical Computing and Communication Technology*, Bangalore, pp. 416-419, 2016.

[20] Ramesh R., Divya G., Divya D., Kurian M., and Vishnuprabha V., "Big Data Sentiment Analysis using Hadoop," *International Journal for Innovative Research in Science and Technology*, vol. 1, no. 11, pp. 92-96, 2015.

[21] Treebank P., https://www.ling.upenn.edu/courses/Fall_2003/ling001/penn_treebank_pos.html, Last Visited 2018.

[22] White T., *Hadoop: The Definitive Guide*, O'Reilly Media, 2015.

Mariam Khader is a PhD Candidate in computer science at Princess Sumaya University for Technology (PSUT), Amman, Jordan. She received the BSc degree in computer networking systems from the World Islamic Science & Education University (WISE) in 2012, Amman, Jordan. She received her MSc Degree in IT security and digital criminology in 2014 from PSUT. Between 2012-2015, she was teacher assistant and then a lecturer at the network department in WISE University. Her interests include digital forensics, network security and big data analytic.

Arafat Awajan is a Full Professor at Princess Sumaya University for Technology (PSUT). He received his PhD degree in Computer Science from the University of Franche-Comte, France in 1987. He has held various administrative and academic positions at the Royal Scientific Society and Princess Sumaya University for Technology. Head of the Department of Computer Science (2000-2003) Head of the Department of Computer Graphics and Animation (2005-2006) Dean of the King Hussein School for Information Technology (2004 - 2007) Director of the Information Technology Center, RSS (2008-2010) Dean of Student Affairs (2011 - 2014) Dean of the King Hussein School for Computing Sciences (2014-2017) He is currently the vice president of the university (PSUT). His research interests include: Natural Language Processing, Arabic Text Mining and Digital Image Processing.

Ghazi Al-Naymat. He received his PhD degree in May 2009 from the School of Information Technologies at The University of Sydney, Australia. He is working as an Associate Professor in the Department of Computer Science, King Hussein School of Computing Sciences at Princess Sumaya University for Technology (PSUT). In addition, he is currently the chair of the computer science department. His research interests include: Data Mining and machine learning, big data, and data science.