A Near Real-Time Approach for Sentiment Analysis Approach Using Arabic Tweets

Anis Zarrad[1*], Izzat Alsmadi[2], Abdulaziz Aljaloud[3]

[1] School of Computer Science, University of Birmingham, Dubai, UAE.
[2] Department of Computing and Cyber Security, University of Texas A&M, San Antonio, TX 77005 USA.
[3] Prince Sultan University, Riyadh, 11586 Saudi Arabia.

* Corresponding author. Tel.: 77142492315; email: a.zarrad@bham.ac.uk
Manuscript submitted September10, 2019; accepted October 8, 2019.
doi: 10.17706/jcp.14.10.596-614

Abstract: Big data storage and real time data analysis are major challenges for IT researchers. The recent massive increase in data has not been accompanied by adequate storage technology and data processing algorithms. Understanding what people think about an idea, a product, a service or a policy is important for individuals, companies and governments. Sentiment analysis process can be used to identify opinions expressed in text on certain subjects. The result accuracy has a direct effect on decision making in both business and government. Our focus in this paper is first to identify the critical issues associated with real-time big data analysis and then to develop a new paradigm on Hadoop Ecosystem with real-time stream data processing to analyze Arabic tweet sentiment on Twitter. To perform real-time analytics, data collection should be performed using Apache Flume in order to move and aggregate all tweets received online (near real-time) to pre-defined locations through a channel called Sinks to the Hadoop distributed file system (HDFS). In addition, due to the serious challenges in Arabic text and speech and the high speed with which tweets arrive, we designed a complex sentiment analysis (SA) module to process each incoming tweet in such a way that no tweets are lost without being analyzed. Also, a sentiment analysis approach to Arabic text was developed using multiple Hive User Defended Functions (UDF). Finally, to guarantee a varied data collection, we proposed a Java MapReduce program for lexicon-based Arabic sentiment analysis, which supports n-gram search in the lexicon. Our approach was applied to determining opinions about MERS virus in the Kingdom of Saudi Arabia on Twitter Public Stream API and the results are discussed.

Key words: Big data, Hadoop, opinion mining, sentiment data analysis, MERS-CoV infection virus, social networks analysis.

1. Introduction

Opinion mining and sentiment analysis are active research trends in natural language processing and data mining. Recently, with the growing importance of social media in many areas such as social science, political science, and business, has created many opportunities for the research community to collect a large amount of data in order to analyze users' opinions, attitudes, and emotions. Conversely, big data has recently become more relevant for storing data beyond the ability of traditional databases [1]. "Big data" is a term used to describe a dataset of massive volume that is complicated to capture, store, manage, and analyze using traditional database and software techniques.

Today, in most enterprise scenarios, the amount of data collected is too great and may exceed the

processing capacity. Big data [2] has the potential to help decision-makers in their businesses to improve operations and make faster, more intelligent decisions for critical situations. Answering the question "What do people think about a specific subject, product, etc.?" has always been a challenge for decision-makers. For example, it is important to know your friends'' opinions about particular device in the market, or service quality offered by specific company. Such opinions may be incorrect and/or untrusted if the quality of collected data is low. Opinions might be positive, negative, or neutral, and sentiment analysis, also known as opinion mining, is the process of identifying positive and negative opinions, emotions, and evaluations [3]. This type of analysis is important if decision-makers are to avoid any misunderstandings and provide valuable results. Timely and accurate information is important to extract relevant meanings for decision making.

The major challenge for an IT manager is to provide big data analysis in near real-time to make better decisions and take meaningful actions at the right time and place. Despite the availability of new tools and technologies such as Hadoop, MapReduce [4], Hive, and Impala [5] for handling large amounts of data at a high rate, however, real time big data analytics lies beyond the ability of typical technologies and software tools.

Our focus in this work is to exhibit the importance of near real-time big data analysis for decision-makers. An efficient architecture is designed and implemented between Hadoop and the social network Twitter to deal effectively in a real-time manner with big data and to collect/store data. The proposed architecture is comprised of four main modules: Data Collector, Arabic Classifier, Semantic Analysis Lexicon Builder, and Hashtag Frequency Finder. A Flume model has been integrated into the big data architecture to control streaming data in the near real-time process and synchronize the data online into the Hadoop distributed file system (HDFS) using a pre-configured agent channel. Having the data in a suitable form for performing appropriate analysis is the most difficult part, especially when dealing with the Arabic language. A Hive User Defined Function (UDF) was developed to perform Arabic sentiment analysis and hashtag frequency analysis. To extend our polarity lexicon, we incorporated a novel semantic similarity analysis (SSA) approach to discover new words of similar polarity from the collected dataset using Apache Hive [6]. A case study of the MERS-CoV infection virus [7] in the Kingdom of Saudi Arabia (KSA) is presented to analyze people's opinions in Saudi Arabia. Twitter API Public Stream is used to collect data and try to answer the following questions related to satisfaction with the available prevention methods proposed by the ministry of health, statistics truthiness, and services offered by the ministry of health. Results can be mapped to different dimensions; for example, we can determine satisfaction and/or non-satisfaction in certain periods of time and certain regions. Fig. 1 shows the main components of the proposed solution.

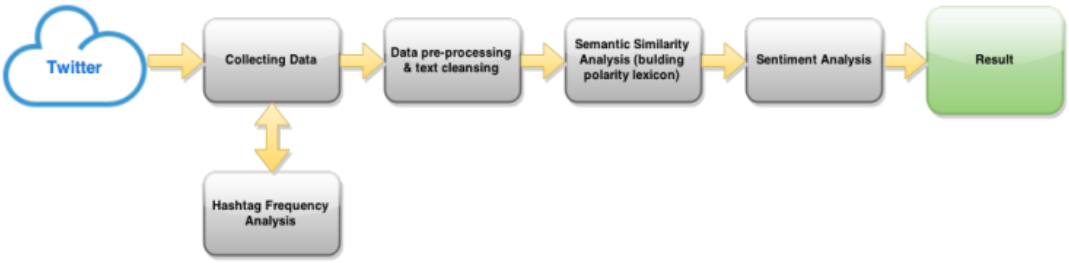

Fig. 1. Proposed system process flow.

We used Twitter as a data source to collect information for this study because Saudi Arabia has the highest percentage of active Twitter users among its online population [8]. In 2012, the Global Web Index reported that 51% of the Saudi Arabia population are active Twitter users; this is give us an indication

about Twitter huge growth usage. Twitter is an online microblogging service that allows users to share their thoughts in 140 characters. A tweet is short and informal, and contains Internet slang words and emoticons. Users tend to express their thoughts and opinions in a straight-forward manner because of the character limit, which increases the chances of achieving high accuracy for opinion mining analysis [9].

The main responsibility of our Data Collector module is to capture any tweets related to our subject in real time, and then synchronize them online into the Hadoop file system. Hashtags on Twitter are words prefixed with '#' that are used to group public messages and discussions [10]. Hashtag frequency analysis is used to extract and count the occurrences for all tagged hashtags in the collected set of tweets. A text pre-processing component is an essential phase for addressing some Arabic language challenges. Arabic language is one of the morphologically rich languages that have significant challenges to Natural Language Processing systems [11]. Various forms can exist for the same Arabic word using different suffixes, affixes and prefixes. Also, there are many varieties of Arabic dialects [12]. Arabic languages used in North Africa, are incomprehensible to an Arabic speaker from the Middle East region. Therefore, using the different dialects in social media, adds more challenging to sentiment analysis because the majority of the existing NLP tools have been developed based on modern standard Arab. It is very difficult to manage the change in polarity classification, when we have various dialects [13]. Researchers are often limited by the Arabic resources available for sentiment analysis.

This paper is organized as follows: Section II provides related works; Section III presents a brief introduction of the proposed architecture and describes the data collection method using Hadoop and the analysis methodology; Section IV analyzes a MERS-CoV case study; and finally Section V concludes the paper.

2. Related Work

In this section, we study previous works related to big data Technologies and Sentiment analysis approaches.

2.1. Big Data Technologies

Technologies such as MapReduce, Hbase, HDFS, Hive, and Pig are used in big data solutions to run queries without changing the underlying data structures.

2.1.1. Hadoop

Hadoop is an open-source Apache-based framework for reliable, scalable distributed computing that allows for the distributed processing of large datasets across clusters of computers using a simple programming model called MapReduce [4], [14], which is used widely by companies such as Yahoo and Facebook [15]. Hadoop is a reliable framework that has been designed to automatically deal with hardware failures. Hadoop MapReduce and HDFS are designed by Google MapReduce and the Google file system [16]. In this work, we use Hadoop as the main component in our architecture.

2.1.2. HDFS

The Hadoop distributed file system (HDFS) is a distributed, scalable, portable file system written in Java for the Hadoop framework [17]. It is an open source implementation of the Google File System (GFS) that is capable of storing and accessing large-scale data-intensive applications within the Hadoop cluster.

2.1.3. Mapreduce

MapReduce is a programming model introduced by Google in 2004 to support distributed and parallel computing on large datasets in Hadoop clusters. Each MapReduce program is composed of two main functions:

- ✓ map(), which does the data processing, sorting, and filtering tasks.
- ✓ reduce(), which aggregates and summarizes all the outputs from the maps.

The MapReduce model [18] is made up of several parts as shown in Fig. 2. First, the input data gets split

into maps, and then the maps carry out the processing in parallel and produce output in key/pair format to the reducers. The reducers then aggregate the output pairs per each key and write the results into files.

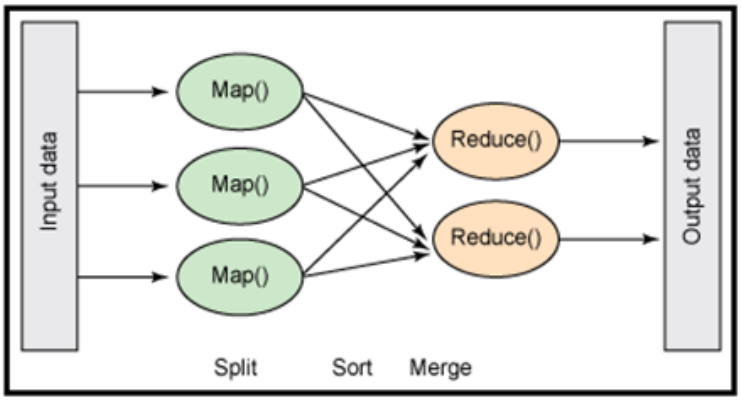

Fig. 2. MapReduce structure [7].

2.1.4. Apache hive

Apache Hive is data warehouse software built on top of Hadoop to analyze and query large datasets stored in Hadoop's HDFS using an SQL-like language called HiveQL [19]. Using Hive, one can easily perform complex queries to retrieve information from big data. In our proposed approach we developed multiple Hive User Defended Functions (UDF), which are used inside Hive queries in order to extract opinions.

2.1.5. Hbase

Hbase is another open source big data technology provided by Apache, and is a distributed NoSQL (non-relational) database that is written in Java programming language and runs within Hadoop and the Hadoop file system. Hbase is designed to manage different forms of large-scale data, specifically petabytes, across nodes in Hadoop clusters. It is reliable and provides a fault-tolerant way of storing data [20].

2.1.6. Apache pig

Apache Pig was developed initially by Yahoo in 2006, and then moved to the Apache Software Foundation [21]. Pig programming language is designed to handle any kind of data. It is a high-level platform that implements a programming language called Pig Latin to create MapReduce programs to be run on top of Hadoop for large-scale dataset analysis. Pig Latin is an easy programming language, and users can extend their programs by implementing UDF for more complex processing purposes [22]. Pig is composed of two main parts: the first is the Pig Latin language and the second is a runtime environment where Pig Latin programs are executed.

2.2. Sentimnet Analysis

In natural language, a given text can be classified into two classes: objective or subjective [9]. Subjectivity refers to aspects of language used to express opinions, evaluations, and speculations, whereas objectivity instead involves facts and concrete bits of information [23].

Sentiment analysis, also known as opinion mining, is the process of classifying the subjective pace of text as positive, negative, or natural [3]. It is one of the most active research areas in natural language processing and is also widely studied in data mining, Web mining, and text mining. Existing sentiment analysis approaches classify subjective text according to the type of input at three levels: document-level, sentence-level and phrasal-level. At document-level the opinion classification is based on the whole document, at sentence-level it is based on whether each sentence in the document expresses a positive, negative, or neutral opinion, and at phrasal-level sentiment analysis tries to classify parts of a sentence [24].

In this work, we will concentrate on sentence-level, as each Tweet is primarily a single sentence. Most of the existing approaches to sentiment analysis are supervised machine learning and unsupervised learning methods. Within supervised learning approaches, there are two types of documents: training set documents, which are the classifier input, and testing set documents for validating the classifier.

Authors in [13] used Twitter API to collect a data set of 8,868 tweets, their goal is to create a reliable sentiment annotated Arabic corpus to support Sensitivity and Sentiment analysis of Arabic text. The annotation for the whole data set was trained with the help of two annotators. Their next future work is to investigate a lexicon-based classifier to automatically annotate the extended data set.

The Naïve Bayes method is one of the easiest and most commonly used methods in supervised machine learning. Naïve Bayes uses text features such as words, bigrams (bags of words), parts of speech (POS), occurrences of polarity words, etc., to classify whether a subjective sentence is holding a sentiment or not [25]. Authors in [26] studied the effects of the variance of pre-processing steps performed before applying different Sentiment Analysis approaches on Arabic text. They collected around 5,500 tweets about sports classes in girls' education in Saudi Arabia. Authors adopted three supervised machine learning methods: SVM, Naïve Bayes and K-Nearest Neighbor (KNN) and then, tested on three pre-processing conditions: no words stemming, using Rapidminer (an open source data mining and machine learning software) Arabic light stemmer and using full Rapidminer Arabic stemmer. The results showed that both the machine learning models of KNN with no stemming and SVM using Rapidminer Arabic stemmer achieved the highest accuracy of 37per cent and 36.96 per cent respectively. There some limitation in this work, the annotation of the data was done manually.

Authors in [27] tried to evaluate customers' preferences on certain products based on social networks analysis and users' comments or posts. Using Hadoop big data, 600,000 Twitter comments from one month period are collected. The proposed architecture includes Hadoop, Twitter4J, HIVE. Many keywords are used to collect the data. Hannanum Java based morphological analyzer is used to process the collected data into sentiments. Three classes in polarity sentiments Positive, negative or neutral are adopted.

Authors in [28] evaluated several classification algorithms used for sentiment analysis in the literature. Authors presented a supervised classification approach to predict the outcome of election result based on users influence in Twitter, their approach is based on Support Vector Machines (SVM), Naive Bayes, Maximum Entropy and Artificial Neural Networks. Two case studies presented: US presidential election 2012 and Karnataka state election in 2013. SVM classification algorithm showed best results in terms of prediction accuracy. Zhang *et al.* [29] defined spam in social networks as irrelevant copied posts or comments. Implemented algorithms detect whether a tweet is a duplicate or not. Authors collected a huge amount of tweets from different users. Users are categorized into 5 classes: Users, robots, information aggregators, marketing accounts and others. Saravanan *et al.* [30] proposed an evaluation approach to investigate the relation between the location and the the tweet itself. Authors in [12] presented a new approach for sentimental analysis using machine learning. A text analysis framework was implemented using Apache spark to build a decision tree for sentimental analysis.

Support vector machine (SVM) is a popular supervised machine learning technique that is a linear classification method. SVM takes a training document set and marks each document into one of two classes or categories (in the case of sentiment analysis, the classes are positive and negative). SVM then builds a model to map a testing document set into its respective class or category [31]. Rule-based method is an event-based supervised learning technique that is commonly used in opinion and knowledge mining. In rule-based method, the text data is molded into a set of condition rules that cause sentiments. In the aspect of event-based analysis, a sentiment can be recalled by a cause event, where the cause events are considered to be the event or condition that triggers the corresponding sentiment [32]. Another supervised

machine learning classification approach proposed in [33]. The goal was to use SVM to extract sentiment from Twitter posts (Tweets). Their presented approach achieved moderate performance on the SemEval sentiment analysis task utilizing. Authors promised an improved performance in their future work by increasing the size of the data set and make a utilization of text normalization.

Unsupervised learning has become more important in recent times with the explosion of social media applications, due to the lack of information labeling and the large amount of data being communicated. The lexicon-based approach is the most popular unsupervised machine learning technique for sentiment analysis [34]. This method is based simply on building a pre-defined or seeded dictionary of polarity words, called a lexicon. This dictionary is used to classify the overall opinion or polarity of a given text. In this work, we incorporate a 5-gram lexicon-based classifier to determine the general sentiment of a given text (i.e., a tweet). In [15] authors presented a semi-supervised machine learning approach to classify Arabic Slang text in Facebook. Also, a slang sentimental lexicon was constructed which contains slang words and idioms used recently in social networks to support the proposed method. The proposed approach consists of three phases: data collection, pre-processing and classification. Authors collected 1846 comments from several data sources like Facebook and news websites.

Banić *et al.* [35] presented a sentiment analysis approach for selecting the most suitable hotel for one's needs. The selection depends on the options of others. Opinions about hotels were collected from the web and evaluated, and these evaluations were aggregated to generate reports for prospective customers and hotel managers. Evaluation was executed offline; hence collected data could have been outdated or unnecessary. Authors in [36] described an innovative big data stream analytics framework named BDSASA, which provided the essential infrastructure to operationalize a probabilistic language modeling approach for near real-time consumer sentiment analysis. The framework contained seven layers. One major limitation related to this approach is that if the sentiment polarity prediction mechanism is not reliable, many words can be missed. Authors in [37] discussed an interesting approach to executing real-time analytics for big data. Data pre-processing could be performed in a way in which only a short summary of the stream is stored in main memory, leading to a reduced processing time and no data being lost without being processed. A vertical Hoeffding decision tree is used to enable parallel classification in distributed environments. Sunil *et al.* [38] developed a sentiment analysis approach using the Hadoop cluster. Faster real-time processing was obtained by using the cluster architecture. The MapReduce program was implemented to process every tweet and assign sentiment to each of the remaining words of a tweet and then sum it up to decide about the overall sentiment.

3. Proposed System Architecture

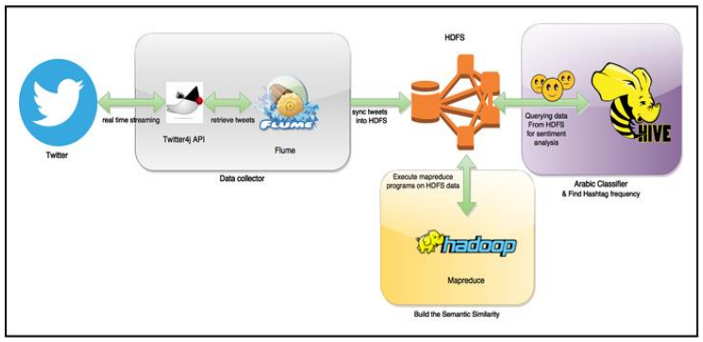

Fig. 3. High-level system architectural view.

This section describes the overall architecture view of the proposed system, and the high-level design of

the main system components and the configuration. The system is composed of four main modules: the data collector, the Arabic classifier, the semantic analysis lexicon builder, and the hashtag frequency finder as shown in Fig. 3. The data collector connects and retrieves tweets and then syncs them online into HDFS. Developed modules then take data from HDFS as an input, carry out the corresponding task, and then write back the results into HDFS again.

In addition, scalable mathematical tools for Semantic Similarity Lexicon, frequency analysis and streaming data analysis approach for complex data sets are discussed in details.

3.1. The Data Collect Component

As an essential step in our approach to building a large dataset and analyzing it in Hadoop, an extremely useful module was designed and implemented in Java. The module is then integrated with Apache Hadoop. The system is designed to collect all Tweets based on search terms related to our MERS-CoV case study. Fig. 4 shows the data collector program using Flume.

Apache Hadoop, Flume version is "a distributed, reliable, and available service for efficiently collecting, aggregating, and moving large amounts of log data" [19]. Our Hadoop will collect all Tweets in real time based on predefined terms to a channel called HDFS Sinks.

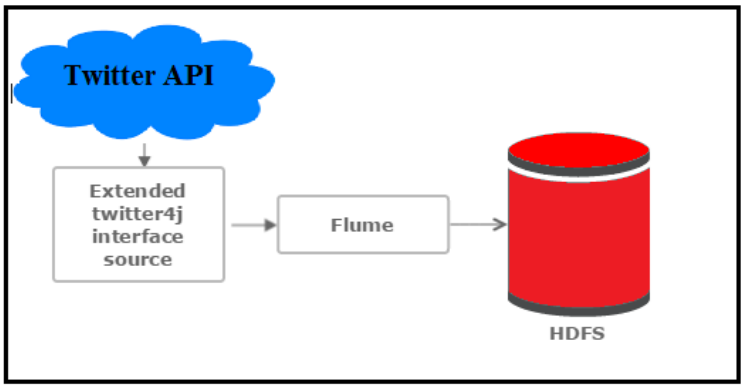

Fig. 4. High-level system architectural view.

We developed a Java program (the Source) which implements integration with Twitter REST API. This is achieved via an open-source Java library called Twitter4j. This library will simplify the complicated integration and authentication processes with Twitter.

The Source is reliable at implemented the capture of all real-time streams, which in the case of this study means that any Tweets posted globally in real time that match our keywords using Twitter Stream API. It is also capable of sifting through old Tweets that were created prior to the first run date or tweets posted after the program gets accidentally stopped by Twitter search API.

Flume played an important role in this module. First, it connects to the Source and starts receiving tweets as events, then Flume moves these events through a memory channel to a memory stage called Sink. All events (tweets) in the sink were aggregated into a 100 MB file each (as per our configuration), then the files were stored in HDFS. This strategy overcomes the delay of storing the data in a local storage and then moving it manually into HDFS.

Flume Agent is a configuration file that brings all parts together (the Source, channel, and the Sink), containing the source arguments, channel type and size, the sink type and the aggregated file size (how many events per file). Table 1 shows an example of the agent configuration file variables and values for the MERS-CoV data collector. Multiple Agent files were created with different variables (keywords set and location) to support and collect different subjects.

Table 1. Flume Agent Configuration File

Variable Name	Value	Description
agent.sources =	Twitter	Source name
agent.channels =	MemChannel	Channel name
agent.sinks =	HDFS	Sink name
agent.sources.Twitter.type =	mse.twitter.flume.TwitterSource	Source class path
agent.sources.Twitter.channels =	MemChannel	Channel used by the source
agent.sources.Twitter.accessTokenSecret=	***	Twitter API OAuth parameter
agent.sinks.HDFS.channel =	MemChannel	Channel used by the sink
agent.sinks.HDFS.type =	hdfs	Sink type
agent.sinks.HDFS.hdfs.path =	hdfs://localhost:9000/user/mse_dataset/MERS/%Y/%m/%d/%H/	Location path to store in HDFS
agent.sinks.HDFS.hdfs.writeFormat =	Text	Write to HDFS format
agent.sinks.HDFS.hdfs.batchSize =	1000	Batch size to write
agent.sinks.HDFS.hdfs.rollCount =	10000	Roll count
agent.sinks.HDFS.hdfs.rollInterval =	600	Roll interval
agent.channels.MemChannel.type =	Memory	Channel type
agent.channels.MemChannel.capacity =	10000	Channel capacity in KB
agent.channels.MemChannel.transactionCapacity =	1000	Channel capacity in transactions.

In our case, we received and stored tweets into HDFS in their original JSON format. JSON is a semi-structured easy-to-read and write data format [40]. Basically, not only the text and date of each tweet were collected. Additionally, all information and metadata coming with Tweets that were archived and not discarded are collected. Those include Tweet's location, number of mentions, number of retweets, hashtags, URLs, etc. Hadoop has the capability to deal with such high volume of data. Such amount of information was kept for different types of analysis in future.

3.2. Building a Semantic Similarity Lexicon

Constructing the polarity lexicon for opinion analysis was the main challenge in our work. Initially, we manually seeded the lexicon by reviewing a collected dataset of related tweets of about 1,100 negative words and 850 positive words. The lexicon contains n-gram words such as: 'حسبي الله عليهم'. We then used an English polarity lexicon constructed by MPQA [23] that contains 8,222 labeled positive, negative, or neutral words or terms. The lexicon was translated by Google Translate into Arabic, and then we did a quick review of the translation result before adding it to our lexicon. We did not investigate more corpora due to the amount of effort required to manually evaluate, translate, review, and adjust sentiment words before adding them to our lexicon.

Table 2 shows a negation list that was constructed. The list contains about 11 words that may emphasize a negative impression when they are combined along with positive words.

Table 2. List of Negation Words

Negation Word List			
غير Non	ليست It is not	ليس Not	ليسوا They are not
لا No	ماهو No	مهو No	ما No
لم Did not	مو No	ماهي No	

We discovered that using the manual lexicon was not very efficient due to the large number of collected words that were not labeled in our lexicon. We then decided to use a mechanism that could extend the

polarity lexicon by analyzing the collected dataset. We implemented a semantic similarity algorithm [39] with an integrated MapReduce program that used pointwise mutual information (PMI) to find the correlation between words in our dataset to discover new polarity words. Semantic similarity measures the degree to which two words are related; as an example, most would agree that wonderful and beautiful are more closely related than are wonderful and awful. In this way, we were able to discover new positive words by finding positive words in the collected tweets (corpus) that were similar to the existing positive words in our manual lexicon.

To describe the algorithm in detail, let us assume W1 and W2 are the two words whose semantic similarity we wish to determine, and C = {c1,c2,...,cm} denotes our corpus of tweets after going through the pre-processing phase, where m is the total words in C. Also, let T = {t1,t2,...,tn} be the set of all unique words in the corpus C. In this work, we set a window size l, which is equal to the length of a tweet. We define the below functions to determine the similarity between W1 and W2 :

Equation 1

$$f^t(t_i) = |\{k: c_k = t_i\}|, where\ i = 1,2,\dots,n$$

Where f^t Where f^t tells us how many times the word t_i occurred in the entire corpus C, and

Equation 2

$$f^b(t_i, W) = \left|\left\{k: t_k = W\ and\ t_{k\pm j} = t_i\right\}\right|, where\ i = 1,2,\dots,n\ and\ 1 \le j \le l$$

where f^b is the bigram frequency function which tells us how many times word t_i is

where f^b is the bigram frequency function which tells us how many times word t_i is followed by the word W in a windows of size l, where l is an even number.

The window is a set of words of size l that contain the word W itself, where $l/2$ is the number of words that precede and succeed the word W.

For every t_i For every t_i having $f^b(t_i, W) > 0$ the PMI function is defined as follows:

Equation 3

$$f^{pmi}(t_i, W) = \log\frac{f^b(t_i, W)\times m}{f^t(t_i)f^t(W)}, where\ i = 1,2,\dots,n\ and\ f^t(t_i)f^t(W) > 0$$

Let X be a set of words in descending order of PMI value with W_1, with the top β_1 Let X be a set of words in descending order of PMI value with W_1, with the top β_1 words having

$f^{pmi}(t_i, W_1) > 0$

Equation 4

$$X = \{X_i\}, where\ i = 1,2,\dots,\beta_1\ and\ f^{pmi}(t_1, W_1) > f^{pmi}(t_2, W_1) > \cdots > f^{pmi}\left(t_{\beta_1\text{-}1}, W_1\right) > f^{pmi}\left(t_{\beta_1}, W_1\right)$$

Similarly, we define Y for W_2 Similarly, we define Y for W_2 as:

Equation 5

$$Y = \{Y_i\}, where\ i = 1,2,\dots,\beta_2\ and\ f^{pmi}(t_1, W_2) > f^{pmi}(t_2, W_2) > \cdots > f^{pmi}\left(t_{\beta_2\text{-}1}, W_2\right) > f^{pmi}\left(t_{\beta_2}, W_2\right)$$

β βis the value that tells us how many times the word W appears in the corpus C. We calculate the value of β as:

Equation 6

$$\beta_i = (\log(f^t(W_1))^2\frac{(\log_2 n)}{\delta}, where\ i = 1,2$$

where n is the number of total unique words in C and δ is a constant value that depends on the size of the corpus. The smaller the corpus we use, the smaller the value of δ we should choose. In our work, we set the value to 6.5.

Now, we calculate the semantically close function $\beta - PMI$ for W_1 and W_2, which which sums the positive

PMI values of all the words that are semantically close to W_2 in Y and are also semantically close to W_1 in X as:

Equation 7

$$f^{\beta}(W_1) = \sum_{i=1}^{\beta_1} (f^{pmi}(X_i, W_2))^{\gamma}, where\ f^{pmi}(X_i, W_2) > 0\ and\ f^{pmi}(X_i, W_1) > 0\ and\ \gamma = 3$$

Last, we calculate the semantic PMI similarity function between two words, W_1 and W_2

Equation 8

$$Similarity(W_1, W_2) = \frac{f^{\beta}(W_1)}{\beta_1} + \frac{f^{\beta}(W_2)}{\beta_2}$$

After performing this algorithm on our collected tweets, the output of this analysis was an undirected edge-weighted graph, where the nodes represented words, the edges connected these nodes if they were similar, and two nodes were connected (similar) if and only if their similarity score was greater than zero.

From the graph, we took the five top-ranked similar nodes (neighbors) by score from each node in the graph. Therefore, a word with many (≥ 3) positive words (from the manual lexicon) within its five neighbors and no existence of negative nodes would become a positive word. The same was also applied for negative – a negative word with many negative neighbors with no existence of positive neighbors became negative. All nodes with almost balanced positive and negative neighbors were classified as a neutral word.

A MapReduce program was designed to be run on Hadoop to perform the classification as following:

- The program took the corpus (our collected dataset after filtering and cleansing the tweets) as inputs.
- The type frequency function f^t for each unique word in the corpus was found.
- The bigram frequency function f^b for each word from Step 1 that appears with any word in a window of size 10 was found.
- The PMI function f^{pmi} for each f^b from Step 2 was found
- β_i values were found and then the summation function $\beta - PMI$ was calculated β-PMI.
- The similarity function was calculated for all words, and the graph was adjusted to have only the 5 top most similar neighbors for each node.
- The list of polarity words, positive and negative, and the graph were taken as inputs.
- The polarity classification was applied for each node and the lexicon was updated.

A Java MapReduce program was developed and designed for this type of analysis. This program applied both the semantic similarity algorithm and the node polarity classification discussed earlier. A MapReduce class FindFB was implemented to find the frequency type and bigram frequency function for all unique words in the dataset, after which it wrote the results into HDFS output files. Find Similarity, as another MapReduce class, used argument output file paths and the list of unique words to find similarities, and then store the output files into HDFS.

3.3. Arabic Sentiment Analyzer and Frequency Analysis Component

In this section, we will show an automatic classifier based on our lexicon that attempts to classify and determine whether a Tweet that is written in Arabic language shows positive, negative, or neutral opinion or emotion. This Tweet can be either modern standard Arabic (MSA) or slang Arabic.

We relied on the classification method proposed in [24], which is based on computing the number of occurrences of positive or negative words in the Tweets, leading to the determination of the overall opinion of the tweet. The analyzer method was extended to support composite words. A tweet is considered to hold an opinion if it contains one or more negative or positive words; otherwise, the tweet is a fact.

It is important to pre-process a tweet's original text to unify the form between words in tweets and the words in the lexicon dictionary before starting the sentiment analysis. Fig. 6 describes the pre-processing approach used to normalize the tweets.

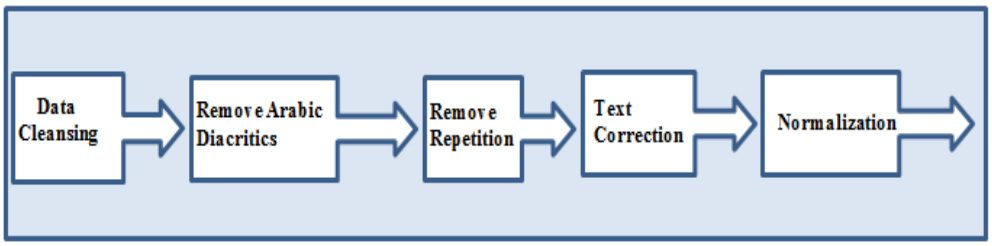

Fig. 5. Pre-processing approach.

Five major steps are required:

- Data Cleansing: In this step, we will remove all unwanted text that is not important for our analysis from tweets, which includes: URLs, @usernames, #hashtags, special characters, white spaces, numbers, and non-Arabic words.
- Remove Arabic Diacritics and Extensions: Some tweets contain diacritics, even though they are not in MSA. In this task, we removed all diacritics and extensions in a tweet.
- Remove Repetition of Letters: Repetition of letters in a word may show emphasis on a certain emotion, for example 'أرجووووووكم كفى' ("Please stop"). This happens frequently on social media, so in this case we try to detect and remove the repeated letters.
- Text Correction: Automating the process of correcting misspelled words is a difficult mission. Hunspell API (www.hunspell.github.io/), an open source spell checker that supports Arabic, was compiled and used. First, every word in a tweet was evaluated by the Hunspell spell check function. If the evaluated word was misspelled, then the function gave probable suggestions for a correction. Finally, a suggested word with the highest probability was selected as the corrected word.

- The Normalization stage: This is a very critical step for unifying the form between words in tweets and the words in the lexicon dictionary. This was carried out via the following steps:
  - o Replacing any occurrence of ى, آ, أ and إ with ا.
  - o Replacing any occurrence of ؤ with و.
  - o Replacing any occurrence of ئ with ي.
  - o Replacing any occurrence of ة with ه.

Table 3 shows an example of pre-processing a tweet and preparing for the final analysis phase. Text pre-processing is an essential phase for addressing some Arabic language challenges, as described in [41].

Table 3. Before and after Tweet Pre-processing Actions

Original tweet	@news_KSA إصاااااااااابةُ 3 أطبـــــاءٍ بمُسْتَشْفى وشَطَ الريـــــــــــاض بفَيروووووس كورونا بَيْنَهُم مَسؤول رَفيع بالمشتشفى. واشتباااااه بإصَابَةِ 3 حالاتٍ أخْرى مخالطة لـــــــهم .. http://dlvr.it/C4tHSSL السعودية#الرياض# *English translation:* Three are three doctors infectedddddd by coronavirus in a hospital downtown Riyadh including a senior official working in the hospital.
Data cleansing	واشتباااااه بإصَابَةِ حالاتٍ أطبـــــاءٍ بمُسْتَشْفى وشَطَ الريـــــــــــاض بفَيروووووس كورونا بَيْنَهُم مَسؤول رَفيع بالمشتشفى إصاااااااااابةُ أخْرى مخالطة لـــــــهم *English translation:* Many Doctors are infectedddddd by coronavirus including a senior official in a hospital.
Remove Arabic diacritics and extensions	واشتباااااه بإصابة حالات أخرى مخالطة أطباء بمستشفى وشط الرياض بفيروووووس كورونا بينهم مسؤول رفيع بالمشتشفى إصاااااااااابة لهم *English translation:* Many Doctors are infectedddddd by coronavirus including a senior official in a hospital.
Remove repetition of letters	واشتباه بإصابة حالات أخرى مخالطة لهم أطباء بمستشفى وشط الرياض بفيروس كورونا بينهم مسؤول رفيع بالمشتشفى إصابة *English translation:* Many Doctors are infected by coronavirus including a senior official in a hospital.

Text correction	واشتباه بإصابة حالات أخرى مخالطة لهم أطباء بمستشفى وسط الرياض بفيروس كورونا بينهم مسؤول رفيع بالمستشفى إصابة *English translation:* Many Doctors are infected by coronavirus including a senior official in a hospital
Normalization	اصابه اطباء بمستشفا وسط الرياض بفيروس كورونا بينهم مسوول رفيع بالمستشفا واشتباه باصابه حالات اخرا مخالطه لهم *English translation:* Many Doctors are infected by coronavirus including a senior official in a hospital.

A tweet was said to be positive if the total number of positive word occurrences was more than the total for negative words, and vice versa. To formalize this, let us assume that t is a tweet and T is the collected training set of tweets, for each t ∈ T:

t is holding an emotion e if POS >0 or NEG >0, otherwise t is a fact.
If POS – NEG is > 0, then e is considered as positive.
If NEG – POS is >0, then e is considered as negative.
If POS is equal to NEG then e is considered as neutral.
Where *e* can be an emotion in {positive, negative or neutral}.
POS is the total number of positive word occurrences in t.
NEG is the total number of negative word occurrences in t.

Algorithm 1 shows how the explained classification method above is implemented. The algorithm is also enhanced to take advantage of Hadoop's processing capabilities. The algorithm searches for up to n-gram words in our polarity lexicon, and can also detect positive words that are affected by a negation word, even if they are not consecutive, as shown in the algorithm.

Algorithm 1 Evaluating Polarity Algorithm

input *is a tweet words* $W = \{w_1, \dots \dots, w_n\}$
Output the *polarity p*
set *pos* ***to*** *0*
set *neg* ***to*** *0*
set *neu* ***to*** *0*
for *i = 1; i <=* ***sizeof****(W)* ***do***
  for *y =* ***min****(****sizeof****(W) – i, 5) ; y >= 1* ***do***
    set $SW = \{w_i, \dots \dots, w_y\}$
    set *e =* ***checkInLexicon****(SW)*
    if*(e = negative)* ***then***
      neg = neg+1
    else if*(e = positive)* ***then***
      set $NW = \{w_{max\,(i\text{-}3,0)}, \dots \dots, w_{i\text{-}1}\}$
      if(isContainNegationWord*(NW) =* ***true)***
      then
        neg = neg-1
      else
        pos = pos+1
      end if
    else
      neu = neu+1
    end if
    y = y-1
  end for
  i = i+1
end for

if*(pos + neg > 0)* ***then***
  p = positive
else if*(pos + neg = 0)* ***then***
  p = netural
else
  p = negative
end if
return *p*

As we mentioned earlier, we applied this analysis to the text of tweets from the collected dataset, which were all stored in their JSON original format. Hive has the capability of interpreting any JSON file format to a data model that can be structured and retrieved by using SQL-like statements.

Hive User Defined Functions (UDF) were developed to perform Arabic sentiment analysis and hashtag frequency analysis, where UDF can be called inside the SQL statements. As an example, the function evaluate(Tweet) implements our algorithm described earlier, which finds the polarity of a tweet text taken as an input, and then returns the polarity as an output.

We were able to perform complex queries on our dataset to extract the polarity of each tweet and join it with different diminutions, such as the creation date of a tweet, then aggregate the results for reporting purposes.

4. System Setting and Expermental Results

4.1. Hadoop Setting and Configuration

Our Hadoop cluster consists of 5 nodes (name node and four data nodes) connected over a 100 MB local LAN. All Hadoop instances are running on virtual machines installed on the connected devices. The overall network cluster is presented in Fig. 6.

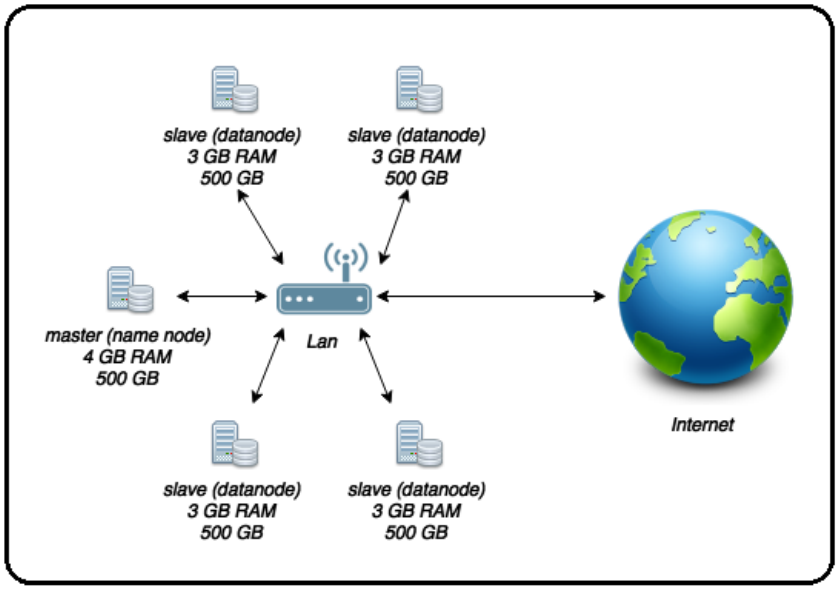

Fig. 6. Diagram of hadoop cluster network.

Different operation systems were installed in each node, and the specifications of the five nodes used for this scenario are presented in Table 4.

Table 4. Specifications of Cluster Nodes

Criteria	Master	Slave 1	Slave 2	Slave 3	Slave 4
CPU	Intel® Core™ i7 2.10 GHz	Intel Core 2 Duo 2.53GHz	Intel® Core™ i5 2.5 GHz	Intel® Core™ i5-3320M 3.30GHz	Intel® Core™ i5 2.4 GHz
Host OS.	Windows 7	Mac 10.8	Windows 7	Windows 7	Ubuntu 12.04
VM OS.	Ubuntu 14.1	Ubuntu 14.1	Ubuntu 14.1	Ubuntu 14.1	Ubuntu 14.1
RAM	4GB	3GB	3GB	3GB	3GB
HDD	500GB	500GB	500GB	500GB	500GB
Network	100baseT/Full	100baseT/Full	100baseT/Full	100baseT/Full	100baseT/Full

This Hadoop cluster was configured with an HDFS capacity of 2000 GB, and all machines were installed with the following software: Ubuntu Linux 14.04, Java 1.7.0, Hadoop version 2.2.0, Hive 0.31.0 (in Master node only), Hue 3.8 (in Master node only), Apache Flume 1.5.0, and Twitter4j 4.0.

4.2. Experimental Results

Over a period of 9 months, from April 2014 to December 2014, we collected approximately 40 million tweets, 2,632,973 of which were related to the targeted topic of MERS-CoV. Table 6 shows some statistics about the training set.

Table 5. Distribution of the Collected Training Set

	Total
#All collected set	~40 million Tweets
#Total related to MERS-Cov	2,632,973 Tweets
#Total Hashtags	6,690,066 Hashtags
#Total unique Hashtags	20,323 Hashtags
#Total processed tokens	103,976,514 tokens

After retrieving tweets from Twitter API, Flume aggregated tweets into 100 MB flat files on HDFS. Hadoop also replicated the same files across all the data nodes in the cluster. Below in Fig. 8 is an example of a subset graph of three similar words (in white, 'أنيق' [elegant],'وذكي' [smart] and 'بتصميم' [design]), showing how are they connected to each other by scores.

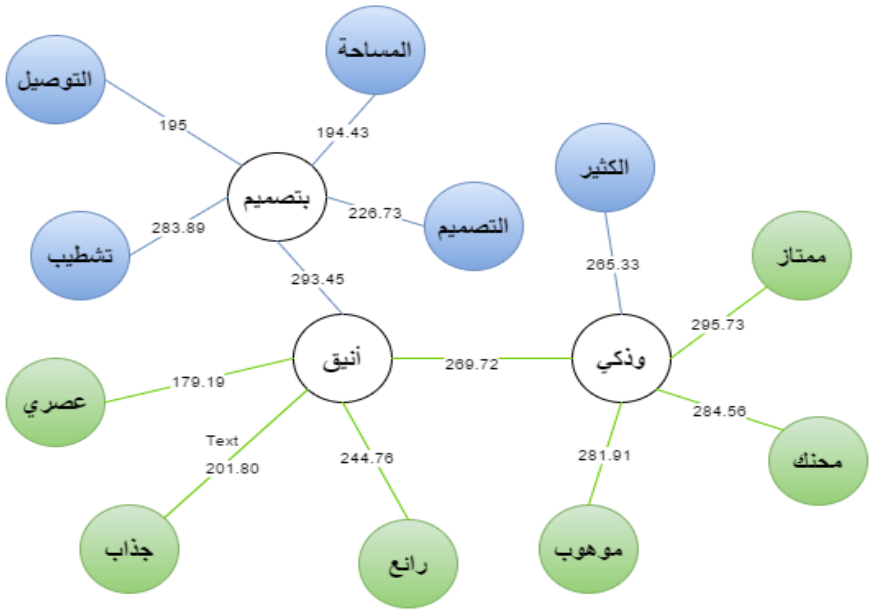

Fig. 7. Word similarity graph for the Words 'أنيق' [elegant],'وذكي' [smart] and 'بتصميم' [design] (shown in white circles).

As can be seen in graph, the word 'وذكي' was connected to three positive words that existed in our seeded lexicon and one not classified (neutral), so this word was automatically added to the positive word list in the lexicon. In the same way, the word 'بتصميم' is connected to only one positive word and four unclassified, so it was labeled as a neutral word (Table 6).

The program found 3,980 similar words connected to the word 'أنيق' ("elegant"). The number of similar words varied from one word to another. The classifier added a wide range of words that had not been included in our initial lexicon, most notably new words, slang, spelling variations, and colloquial emotions.

Table 6. Examples of Positive and Negative Tweets about MERS-CoV Classified by Our Algorithm

Date	Tweet's text	Opinion
2014/07	منظمة الصحة العالمية تؤكد سيطرة المملكة على كورونا بحسب صحيفة الشرق الأوسط.. لاشك انها بحمد الله اخبار ممتازة شكرًا وزارة الصحة #السعودية *English translation:* According to the Middle East newspaper, the world health organization confirms the control of the Saudi Arabia Kingdom of the Coronavirus. There is no doubt it is the praise of God and excellent news, thanks the Ministry of Health Saudi Arabia	Positive
2014/04	منظمة الصحة العالميه : الصحه بالسعوديه سيئة وغير قادره على مواجهة كورونا وسنتدخل قريبا حتى لا ينتشر الوباء للعالم اجمع *English translation:* World Health organization: the health situation in Saudi Arabia is bad and unable to cope with the coronavirus; we will intervene soon to stop the spread of the epidemic to the whole world	Negative

Arabic sentiment analysis was carried out via Apache Hive, with easy to complex queries submitted to the Hive engine. Hive then executed the queries by converting them into MapReduce model jobs. Finally, the output results were exported into different locations inside HDFS for either intermediate stages or final presentations.

After performing the Arabic sentiment analysis on all the collected tweets that were related to our case study, the results showed that 25.01 per cent of the tweets were positive, 20.02 per cent were negative, and 54.97 per cent were neutral. Results are shown in Fig. 8.

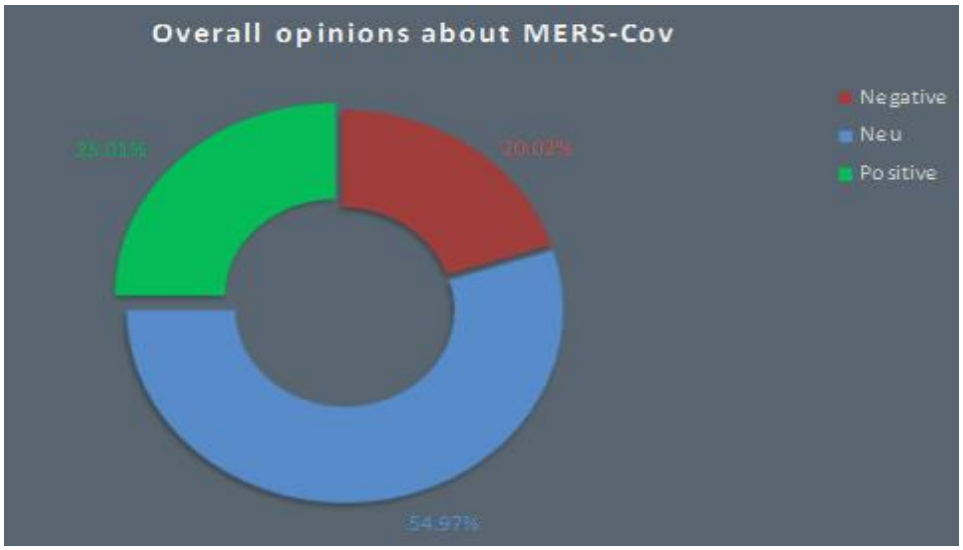

Fig. 8. Overall opinions about MERS-CoV.

There was a noticeable increase in positive tweets in June and July, along with a decrease in the total actual reported cases in the same months shown in Fig. 9, which may explain satisfaction during those two months.

We also applied the time dimension to the hashtag '#فيروس_كورونا' as an example, to measure the activity of the hashtag during the months from April until December as detailed in Fig. 10.

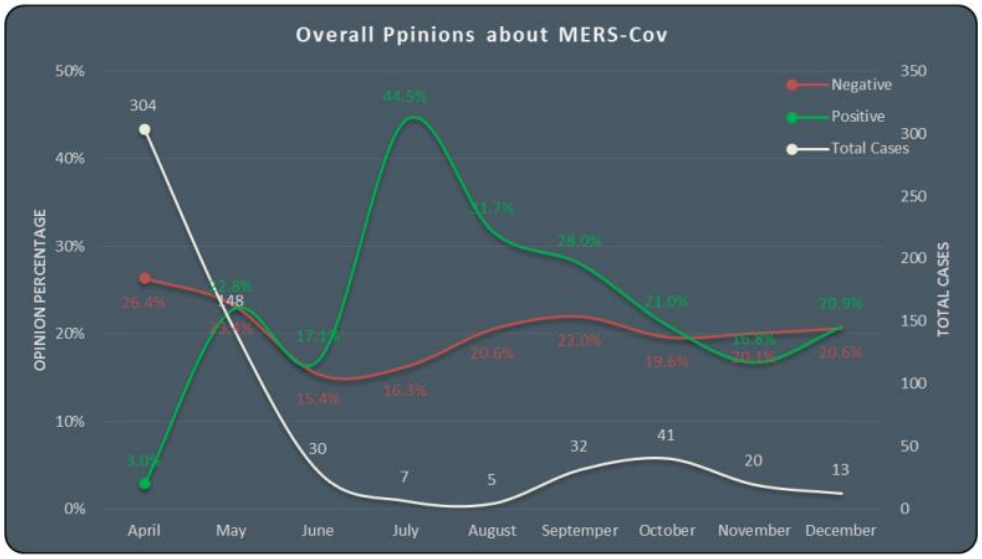

Fig. 9. Overall opinions about MERS-CoV from April, May, to December 2014.

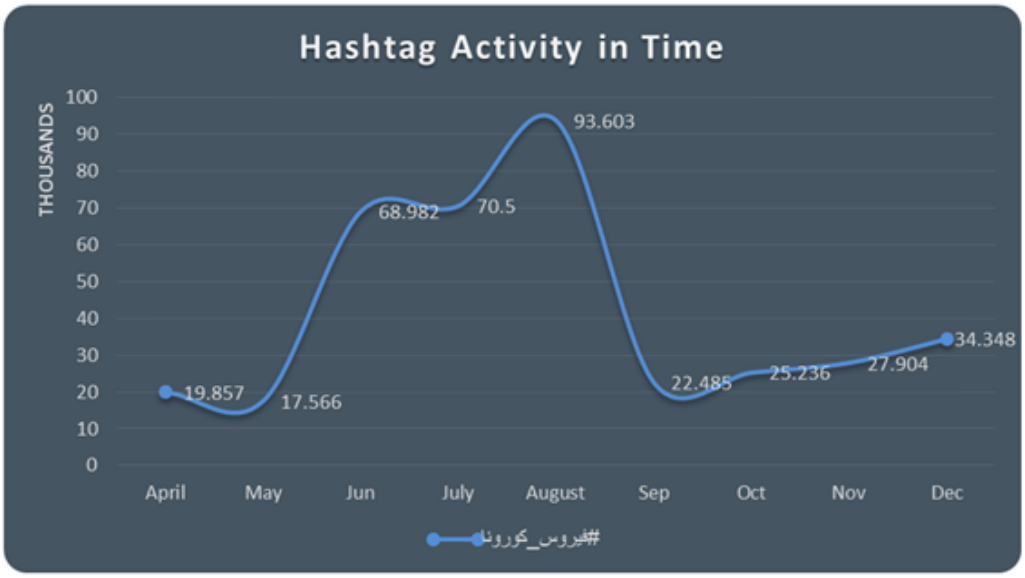

Fig. 10. Activity of the hashtag #فيروس_كورونا during April through December, 2014.

From the hashtag frequency analysis, we determined the opinion polarity for two of the top 33 most frequently-used hashtags.

5. Conclusion

The value of information collected from social networks has been the subject of many studies in different fields, which have combined different technology and social contexts. In this paper, we proposed a system for collecting and analyzing large-scale social data. The developed data collection module was able to connect to Twitter API and retrieve, aggregate, and synchronize tweets into HDFS online. A dataset of around 40 million tweets was collected and analyzed. A hashtag frequency analysis was performed to support the data collection module in exploring and discovering more search keywords to help maximize the amount of data collected. This analysis resulted in a list of top-ranked search keywords that related to the main subject. A semantic similarity analysis was also carried out to analyze the collected data with the purpose of extending the manually constructed polarity lexicon. This analysis helped by labeling polarity words from the dataset that had not existed in the lexicon; approximately 6,000 polarity words were added

to the lexicon. We proposed and enhanced a lexicon-based Arabic sentiment analysis which supported n-gram search in the lexicon.

Overall, results were promising, which motivates us to continue working on this subject. We plan to provide an enhanced system by using hashtag frequency analysis to extend the set of search keywords during the collecting of tweets and by automating the process of using semantic similarity to extend the polarity lexicon. Furthermore, we will investigate more existing lexicon corpora in Arabic or English to extend and enhance our lexicon. In addition, we plan to enhance sentiment analysis by improving the search capabilities for lexicon words and improving the mechanisms of negation and emoticon extraction.

References

[1] Gantz, J., Reinsel, D., Chute, C., Schlichting, W., McArthur, J., Minton, S., Xheneti, I., Toncheva, A., & Manfrediz, A. (2007). The expanding digital universe. *IDC White Paper, Sponsored by EMC.*

[2] Manyika, J., Chui, M., Brown, B., Bughin, J., Dobbs, R., Roxburgh, C., & Byers, A. (2011). *Big Data: The Next Frontier for Innovation, Competition, and Productivity.* Report McKinsey Global Institute.

[3] Tchalakova, M. (2007). Identifying positive and negative expressions — A step towards more sensitive eLearning systems. *EUROLAN.*

[4] Prasad, M., Mrigank, R., Khajuria, A., Snehasish, D., & Kumar, N. (2014). Analysis of big data using Apache Hadoop and map reduce. *International Journal of Advanced Research in Computer Science and Software Engineering, 4(5)*, 145-166.

[5] Jingmin, L. (2014). Design of real-time data analysis system based on Impala. *IEEE Workshop on Advanced Research and Technology in Industry Applications (WARTIA)* (pp. 934-936). Canada.

[6] Thusoo, J., Sarma, S., Jain, N., Shao, Z., Chakka, P., Anthony, S., Liu, H., Wyckoff, P., & Murthy, R. (2009). Hive – A warehousing solution over a map- Reduce framework. *Proceedings of the VLDB Endowment, 2(2),* 1626-1629.

[7] Zarrad, A., Jaloud, A., & Alsmadi, I. (2014). The evaluation of the public opinion — A case study: MERS-CoV Infection Virus in KSA. *UCC,* 664-670.

[8] Mari, M. (2012). Twitter usage is booming in Saudi Arabia. *Global Web Index*.

[9] Pang, B., & Lillian, L. (2008). Opinion mining and sentiment analysis. *Journal Foundations and Trends in Information Retrieval*, *2(2),* 135-146.

[10] Oren, T., & Ari, R. (2012). What's in a Hashtag? Content based prediction of the spread of ideas in microblogging communities. *Proceedings of the fifth ACM International Conference on Web Search and Data Mining* (pp. 643-652). USA.

[11] Abdul-Mageed, M., Diab, T., & Korayem, M. (2011). Subjectivity and sentiment analysis of modern standard Arabic. *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics* (pp. 587-591). USA.

[12] Zaidan, F., & Callison-Burch, C. (2011). The Arabic online commentary dataset: An annotated dataset of informal Arabic with high dialectal content. *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics* (pp. 37-41). USA.

[13] Refaee, E., & Rieser, V. (2014). Arabic twitter corpus for subjectivity and sentiment analysis. *Proceedings of the Ninth International Conference on Language Resources and Evaluation* (pp. 26-31). Iceland.

[14] Karun, K., & Chitharanjan, K. (2013). A review on hadoop — HDFS infrastructure extensions. *Information & Communication Technologies (ICT),* 113-121.

[15] Soliman, T., Elmasry, M., Hedar, A., & Doss, M. (2014). Sentiment analysis of Arabic slang comments on Facebook. *International Journal of Computers & Technology*, *12(5),* 387-398.

[16] Sanjay, R. (2013). Big data and Hadoop with components like flume, pig, hive and Jaql. *Proceedings of International Conference on Cloud, Big Data and Trust* (pp. 1-13).

[17] Shafer, J., Rixner, S., & Alan, L. (2010). The Hadoop distributed filesystem: Balancing portability and performance. *Proceedings of IEEE International Symposium on Performance Analysis of Systems & Software (ISPASS)* (pp. 202-213).

[18] Clemente-Castelló, F., Nicolae, B., Mayo, R., Juan Carlos, F., & Loreti, D. (2015). Enabling big data analytics in the hybrid cloud using iterative MapReduce. *Proceedings of IEEE/ACM 8th International Conference on Utility and Cloud Computing* (pp. 290-299).

[19] Sanjay, G., Howard, G., & Shun-Tak, L. (2003). The google file system. *ACM Symposium on Operating System Principles* (pp. 213-221).

[20] Das, T., & Kumar, P. (2013). Big data analytics: A framework for unstructured data analysis. *International Journal of Engineering and Technology, 5(1),* 135-156.

[21] Olston, C., Reed, B., Srivastava, U., Kumar, R., & Tomkins, A. (2008). Pig Latin: A not-so-foreign language for data processing. *Proceedings of the ACM SIGMOD International Conference on Management of Data* (pp. 1099-1110).

[22] Ammar, F., Alva, E., & Heru, P. (2014). Processing performance on Apache pig, Apache hive and MySQL cluster. *Proceedings of International Conference on Information, Communication Technology and System* (pp. 53-67).

[23] Wiebe, J., Wilson, T., Bruce, M. B., & Martin, M. (2004). Learning subjective language. *Journal Computational Linguistics, 30(3),* 277-308.

[24] Mohammed, N., Al-Kabi, N., Alsmadi, I., Gigieh, H., & Wahsheh, A. (2014). Opinion mining and analysis for Arabic language. *International Journal of Advanced Computer Science and Applications*, *5(5),* 181-195.

[25] Ravi, K., & Ravi, V. (2015). A survey on sentiment analysis algorithms for opinion mining. *Journal Knowledge-Based Systems, 89(C),* 14-46.

[26] Harbil, A., & Emam, A. (2015). Effect of Saudi dialect processing on Arabic sentiment analysis. *International Journal of Advanced Computer Technology, 5(2),* 236-258.

[27] Kim, S., Yang, H., Hwang, Y., Jeon, J., Kim, Y., Jung, I., Choi, H., Cho, S., & Na, J. (2012). Customer preference analysis based on SNS data. *Proceedings of Second International Conference on Cloud and Green Computing* (pp. 106-113).

[28] Anjaria, M., & Guddeti, R. (2014). Influence factor based opinion mining of twitter data using supervised learning. *Proceedings of the 6th International Conference on Communication Systems and Networks COMSNETS,* (pp. 1-8).

[29] Zhang, Q., Ma, H., Qian, W., & Zhou, A. (2013). Duplicate detection for identifying social spam in microblogs. *Proceedings of IEEE International Congress on Big Data* (pp. 52-61).

[30] Saravanan, M., Sundar, D., & Kumaresh, S. (2001). Probing of geospatial stream data to report disorientation. *IEEE Recent Advances in Intelligent Computational Systems (RAICS), 4,* 87-95.

[31] Khan, A., Atique, M., & Thakare, V. (2015). Sentiment analysis using support vector machine. *International Journal of Advanced Research in Computer Science and Software Engineering, 5(4),* 125-136.

[32] Gao, K., Xu, H., & Wang, J. (2015). A rule-based approach to emotion cause detection for Chinese micro-blogs. *Expert Systems with Applications, 42(9),* 4517-4528.

[33] Sarker, A., Nikfarjam, A., Weissenbacher, D., & Gonzalez, G. (2015). DIEGOLab: An approach for message-level sentiment classification in twitter. *Proceedings of the 9th International Workshop on Semantic Evaluation* (pp. 510-514). USA.

[34] Hu, X., Tang, J., Gao, H., & Liu, H. (2013). Unsupervised sentiment analysis with emotional signals. *Proceedings of the 22nd international conference on World Wide Web* (pp. 607-618).

[35] Banić, L., Mihanović, A., & Brakus, M. (2013). Using big data and sentiment analysis in product evaluation. *Information & Communication Technology Electronics & Microelectronics (MIPRO),* 1149-1154.

[36] Otto, K., Cheng, M., & Raymond, L. (2015). Big data stream analytics for near real-time sentiment analysis. *Journal of Computer and Communications, 15(3),* 189-195.

[37] Amir, H., & Akhavan, R. (2014). Distributed real-time sentiment analysis for big data social streams.

Proceedings of International Conference Control, Decision and Information Technologies (pp. 789-794).

[38] Sunil, M., Yashwant, S., Kazi, S., & Vaibhav, S. (2014). Real time sentiment analysis of twitter data using Hadoop. *International Journal of Computer Science and Information Technologies, 5(3),* 3098-3100.

[39] Aminul, I., & Inkpen, D. (2006). Second order co-occurrence PMI for determining the semantic similarity of words. *Proceedings of the International Conference on Language Resources and Evaluation*, (pp. 1033-1038).

[40] Mesiti, M., & Valtolina, S. (2014). Towards a user-friendly loading system for the analysis of big data in the internet of things. *Proceedings of International Conference Computer Software and Applications* (pp. 203-214).

[41] Ali, F., & Khalid, S. (2009). Arabic natural language processing: Challenges and solutions. *ACM Transactions on Asian Language Information Processing, 4,* 14-36, 2009.

Anis Zarrad is a lecturer at the Computer Science School in the University of Birmingham. He received his PhD in computer science -University of Ottawa, Canada. Dr. Anis has a master degree in computer science from Concordia University, Canada. His research interest are big data analysis, peer-to-peer networks, software testing, and mobile collaborative virtual environment to model emergency preparedness scenarios. Dr. Anis has published several research papers in conferences and international journals.

Izzat Alsmadi is an assistant professor in the department of computing and cyber-security at University of Texas A&M San Antonio. He obtained his Ph.D degree in software engineering from NDSU (USA) in 2008. He has two master degrees in software engineering and CIS from NDSU (USA) and University of Phoenix (USA). He has several published books, journals and conference articles largely in software engineering, information/computer security and information retrieval.

Abdul-Aziz Aljaloud is a master student in software engineering program in the department of computing science and information systems at Prince Sultan University. His research interests include big data, sentimental analysis, and software testing.