Discovering and Analyzing Important Real-Time Trends in Noisy Twitter Streams

Khalid N. Alhayyan, Assistant Professor
Information Technology Department, Institute of Public Administration,
P.O. Box 205, Riyadh 11141, Saudi Arabia
alhayyan@ipa.edu.sa

and

Imran Ahmad, Assistant Professor
Data Scientist, Cloudanum Inc.,
22 Castlebrook Lane, Canada, ON, K2G 5G2
imran@cloudanum.com

ABSTRACT

We present an approach, called StreamSensing, suitable for processing real-time data in noisy streams. This approach consists of six stages: (1) tokenization, (2) stop words removal, (3) stemming, (4) filtering, (5) conversion into Term Document Matrix (TDM), and (6) pattern analysis. The approach was experimentally tested and implemented using a fast in-memory processing system, called Spark. The results of such implementation are reported and analyzed. The findings of this paper fall into two perspectives: theoretical and practical. The theoretical perspective is represented in the introduction of the StreamSensing approach, while practically; this approach can be employed to perform trend analysis on any real-time text data stream.

Keywords: StreamSensing, Real-Time Trends, Noisy Stream, Trend Analysis and Pattern Analysis.

1. INTRODUCTION

In the past, traditional media, such as TV, radio, newspapers, or magazines used to dominate the globe as the source of reporting about events. Recently, real-time social media, such as Facebook, Twitter, Flicker, YouTube, and Instagram have become a significant tool of spreading emerging news (Kwak et al., 2010), even before traditional media can confirm and report on the news. The more the news travel across the social media, say Twitter, the more likely users of social media participate in the same topic, contributing to the trending of a topic (Johnson 2009; Popescu and Pennacchiott, 2011). Trending topics are those topics being discussed more than others are. As Twitter Inc. explains trending topics, "Twitter Trends are automatically generated by an algorithm that attempts to identify topics that are being talked about more right now than they were previously." These trending topics can refer to real world events such as political movements (USA election), financial events (interest rate hike), product releases (iPhone 7), and entertainment (Academy Awards). Awareness of trending topics plays a key role in building social satisfying users' information needs. Noisy Twitter streams refer to an unordered huge number of multi-topic and unfiltered tweets that come as an input at very high rate, so it is hard to transmit, compute, and store. In many ways, these streams reflect changing interests of groups of individuals. Therefore, Twitter streams mirror our society to a significant degree. These streams contain rich information and immediate feedback about what people currently pay attention to and how they feel about certain topics. (Schubert et al., 2014, Althoff et al., 2013).

Data mining techniques have been used in various research fields, such as marketing, medicine, and sociology, for many years and have been proved to be very effective. While extracting data patterns is a goal for this study, a good deal of effort needs to be exerted at understanding how such employed data is structured, distributed, and related among its components. Specifically, we focus in this paper on discovering and analyzing real-time trends in real-time social media streams. These trends can be used to discover emerging patterns in real time, which can be used for various applications, including predicting the performance of financial markets (Bollen et al., 2011; Mittal et al., 2012), identifying relevant events (Becker et al., 2011), building content-based recommender systems (Chen et al., 2010), detecting emerging security threats (Fire et al., 2014), and improving decision making and business intelligence (Farzindar 2012). Driven by the interest to harvest social media real-time data, there is lots of interest in processing and finding interesting patterns in live streams of social media data. Twitter has become one of the most popular social media technology, which is driven by short messages called tweets, which are used for the sake of information exchange and communications. Currently around 6500 tweets are published per second (source: Twitter Inc.), which results in approximately 561.6 million tweets per day. This huge amount of data posted daily is deemed to contain a wealth of information, and possible data patterns implying sort of useful trends in a specific subject. On the other hand, this live streams of social media data may bring with it number of challenges that needs to be considered when it is offered for real-time pattern detection and analytics. Examples of which includes the challenges of processing unstructured data, increasing signals to noise, and high rates of arriving data. To process such high velocity real-time data, efficient in-memory distributed processing systems are needed to satisfy the processing needs. In this paper, we use Apache Spark to extract keywords from Twitter streams and perform unsupervised learning operations on them for pattern discovery and analysis.

As with any research work, this paper contains a section of related works and literature reviews. It also introduces some notations to formally represent the Twitter stream in section 3. The paper introduces, as well, the StreamSensing approach that explains how keywords in tweets are extracted due to its importance for detecting the real-world events from social media. Additionally, the paper contains sections for experimental setup, analysis and results, and conclusion.

2. LITERATURE REVIEWS

In this paper, we divided our efforts on conducting the literature reviews into two main categories. First, we aimed these efforts to the broad area of mining real-time data streams. Second, we consider the specific area of discovering and analyzing real-time streams in Twitter.

Data stream mining is considerably different from traditional data mining with respect to: (1) the higher rate of arriving data associated with the data stream, and (2) the necessity of maintaining a quick response time to queries on such data streams that are expected to highly utilize computing resources: high CPU overhead and fast in-memory processing (Reddy et al., 2014). Considering these challenges, it would not be acceptable to accommodate traditional data mining technologies on real-time data streams. Therefore, contemporary research has shifted the gears to appropriately find and employ new mining technologies that reasonably fit the needs of mining live streams of data. In general, mining real-time data streams may involve one of two approaches: summarization, or looking at a time window of a stream. Summarization involves selecting a useful sample of a stream, filtering the stream to eliminate most of undesirable elements, estimating the number of different elements, and then introducing these chosen elements for mining (Muthukrishnan, 2005). On the other hand, Zhu et al. (2002) suggest a stream data processing model based on selecting a time window of the stream. Based on this model, mining may be applied through one of three methods: (1) landmark-window based mining, (2) damped-window based mining, and (3) sliding-window based mining (Li et al, 2009; Giannella et al. 2004). A landmark-window model considers the data in the data stream from the beginning of a landmark time until now. So, users of this model are interested in the historical data starting from a user-defined landmark time. A damped-window model considers all data in the data stream but it assigns heavier weights towards recent data than those in the past. The challenge in these two models is in the window size that increases continuously as time progresses. A sliding window model, on the other hand, considers the data from now down to a certain range in the past. Therefore, the targeted data is within limits of a fixed-size time window of the most recently streamed data.

We now consider the specific area of discovering and analyzing real-time streams in Twitter. Doing so brings a necessity to shed some light on what distinguishes Twitter streams from other streams. In that, Twitter messages are restricted in length and written by anyone, while most media messages are well written, structured, and edited. Therefore, tweets may include large amounts of informal, irregular, and abbreviated words, large number of spelling and grammatical errors, and improper sentence structures and mixed languages. In addition, Twitter streams contain large amounts of meaningless messages (Hurlock and Wilson 2011), polluted content (Lee et al. 2011), and rumors (Castillo et al. 2011), which negatively affect the performance of the detection algorithms (Atefeh et al., 2013). Prior researches have proposed various techniques for Twitter stream discovery. Depending on the discovery method, the presented techniques can be categorized into supervised and unsupervised (or a combination of both) techniques. Most techniques for noisy Twitter streams rely on clustering approaches, which are naturally suitable for because they are unsupervised in that they require no labeled data for training. However, these clustering approaches must be efficient and highly scalable, and they should not require any prior knowledge such as the number of clusters (Atefeh et al., 2013). Three recent research works have employed such unsupervised clustering techniques for detecting Twitter streams. First, Becker et al. (2011a) focused on online identification of real-world event content and its associated Twitter messages using an online clustering technique, which continuously clusters similar tweets and then classifies the clusters content into real-world events or nonevents. Second, Cordeiro (2012) proposed a continuous wavelet transformation based on hashtag occurrences combined with a topic model inference using Latent Dirichlet Allocation (LDA). Instead of individual words, hashtags are used for building wavelet signals. A spike increase in the number of a given hashtag is considered a good indicator of an event that is happening at a given time. Third, Long et al. (2011) adapted a traditional clustering approach by integrating some specific features to the characteristics of microblog data. These features are based on "topical words," which are more popular than others with respect to an event. Topical words are extracted from daily messages based on word frequency, word occurrence in a hashtag, and word entropy. A (top-down) hierarchical divisive clustering approach is employed on a co-occurrence graph (connecting messages in which topical words co-occur) to divide topical words into event clusters.

Drawing upon these research efforts and synthesizing the different approaches and techniques employed for analysis and pattern discovery, this paper proposes a staged approach, appropriate for analyzing and discovering real-time noisy streams, called StreamSensing. This approach consists of six stages: (1) tokenization, (2) stop words removal, (3) stemming, (4) filtering, (5) conversion into Term Document Matrix (TDM), and finally (6) pattern analysis. While StreamSensing considers preparing data for analysis in steps from 1 to 5, it avoids employing clustering techniques in step 6 for scalability and efficiency reasons where they incur more time delays necessary for creating and maintaining clusters. To experimentally test the StreamSensing approach, a fast in-memory processing system, called Spark, which is capable of processing high rate of incoming streams, was employed to implement our proposed approach, and the results of such implementation are reported.

3. NOTATIONS

A collection of related symbols is presented in this section to facilitate the communication of the concepts employed in this paper. Therefore, a tweet i from a particular twitter stream is represented as tw_i, consisting of a set of words W_i. A tweet is identified by id_i, and labeled by its creation time $time_i$. We assume that the components (id_i, W_i, $time_i$) exist for each tweet in the stream under consideration. To deal with huge amount of incoming tweets, a sliding window is enforced to incrementally process the recently posted tweets. For that, the timeline is split into fixed-length time intervals called snapshots (..., t-2 ,t-1 ,t), where t is the current snapshot.

4. METHODOLOGY

Twitter streams generate real-time high velocity data. Every second, on average, around 6,500 tweets are tweeted, which corresponds to around 390,000 tweets sent per minute. There is a public API through which tweets are available to researchers through a public streaming APIs, which provides a continuous stream of tweets. Twitter APIs can be accessed only via

authenticated requests and each request must be signed with valid Twitter user credentials. Access to Twitter APIs is also limited to a specific number of requests within a time window called the rate limit. The authentication and authorization of researchers is carried out using Open Authentication (OAuth) which is an open standard for authentication, adopted by Twitter to provide access to protected information. Requests to the APIs contain parameters such as hashtags, keywords, geographic regions, and Twitter user IDs. Responses from Twitter APIs are sent in JavaScript Object Notation (JSON) format, which is a popular format that is widely used as an object notation on the web. A Twitter user's tweets are also known as status messages. A tweet can be at most 140 characters in length. A user's tweets can be retrieved using the Streaming API.

Our approach, StreamSensing, for discovering and analyzing real-time trends in Twitter streams implements the sliding window mechanism where the targeted data is within limits of a fixed-size time window of the most recently streamed data. The determination of the sliding window size (i.e. 5 minutes) depends on many factors such as the dynamism of the targeted topic, the expected overhead of utilizing the computing resources, and the rate of arriving data stream. Each tweet taken from a window is denoted (id_i, W_i, $time_i$), where id_i refers to the tweet identification, W_i refers to the number of words in the tweet i, and $time_i$ refers to the tweet's creation time. The contents of each window then pass through six stages for reaching the pattern analysis. These stages are tokenization, stop words removal, stemming, filtering, conversion into Term Document Matrix (TDM), and finally pattern analysis. Figure 1 shows Twitter stream windowed into set of snapshots (Window A and Window B), and shows the methodology process flow represented by six stages. The incoming streams are first tokenized into list of tokens (i.e. words). Then, stop words, such as "is" and "the", are removed from the token list. The remaining tokens are then reduced to their stems or roots, and then the filtering phase takes place for filtering out the unnecessary characters such as $, @, or #. The filtered stream is then converted into a structured called TDM, which represents the terms and frequency of each word in the corpus, in a structured way. The final stage is the pattern analysis for finding the most important keywords using the TF-IDF statistics (Wu etal., 2008). Table 1 presents the six stages explained with their definitions. Typically, the TF-IDF weight is composed by two terms. The first term computes the normalized Term Frequency (TF) by dividing the number of times a word appears in a document (i.e. tweet) by the total number of words in that document. The second term is the Inverse Document Frequency (IDF), computed as the logarithm of the number of the documents in the corpus divided by the number of documents where the specific term appears. So, TF measures how frequently a term occurs in a document. Since every document is different in length, it is possible that a term would appear much more times in long documents than shorter ones. Thus, the term frequency is often divided by the document length (the total number of terms in the document) as a way of normalization, see Eq. (1).

$$TF(t) = \frac{\text{Number of times term } t \text{ appears in a document}}{\text{Total number of terms in the document}} \quad (1)$$

IDF measures how important a term is. While computing TF, all terms are considered equally important. However it is known that certain terms, such as "is", "of", and "that", may appear many times but have little importance. Thus, we need to weigh down the frequent terms while scale up the rare ones, by computing the formula in Eq. (2).

$$IDF(t) = \log \frac{\text{Total number of documents}}{\text{Number of documents with term t in it}} \quad (2)$$

Let us consider a tweet containing 35 words wherein the word 'man' appears 5 times. The term frequency (i.e., TF) for 'man' is then (5 / 35) = 0.143. Now, assume we have 10 million tweets and the word 'man' appears in only 1000 of them. Then, the inverse document frequency (i.e., IDF) is calculated as log (10^7/ 10^3) = 4. Thus, the TF-IDF weight is the product of these quantities: 0.143 * 4 = 0.572.

The mechanism that implements our approach requires that we first choose a trending topic for analysis. Then Spark, an automated system, is engaged with the chosen hashtag and starts capturing the real-time data associated with this hashtag. Capturing real-time data is a continuous process, however, since we implement the sliding window model, real-time data collection is controlled. That means, if we assume that we have the sliding window size set to 5 minutes, then data is captured each minute within this 5-minute window. The sliding window moves ahead after each minute, and we have one minute to accomplish all the necessary processing (analysis and aggregation) and collect the results before new set of tweets are collected. Results should show a set of keywords, selected from the chosen hashtag, listed in order based on their TF-IDF values. For example, if we choose the hashtag #Ottawa, then the twitter stream is represented as $tw_{\#Ottawa}$. A number of n tweets is captured each minute for this particular stream. Each tweet i within these n tweets has the following three components. (id_i, W_i, $time_i$). A TDM (represented as $TDM_{\#Ottawa,t}$) is created, and it summarizes the occurrences of Tweeter stream at this instance of time t. From each individual TDM, updated after each minute, top 20 words are chosen and further analyzed. For all of those 20 words, top five words, displaying highest value of TF-IDF, are selected and considered as the words depict the current trends in the chosen hashtag of the twitter stream.

5. EXPERIMENTAL SETUP

The data for the experiment was gathered in two different time-slots. The first experiment, lasted for 40 minutes, was conducted on Dec 27, 2016 at 8:00 pm EST using two hashtags: #BigData and #NewYork. Then on May 6, 2017, the second experiment, lasted for 45 minutes, was conducted at 7:00 pm EST using one hashtag: #Ottawa. The choice of the hashtags #NewYork and #Ottawa was due to their Twitter trending during the experiment conduct, while #BigData was chosen for its low popularity during the experiment conduct. Selecting different levels of popularity is to ensure that our proposed approach, StreamSensing, can scale and handle different levels. For the expected high level of results' dynamism pertaining to the chosen hashtags during the experiment conduct, the sliding window size was set to five minutes for both experiments.

A cluster of five computers was organized for running Spark streaming. Spark has emerged as the next generation of big data processing engine, overtaking Hadoop MapReduce, which originally helped, ignite the big data revolution. Apache Spark

maintains MapReduce's linear scalability and fault tolerance, but extends it further. Depending on the type of the application, Spark can process 100 times or faster than the traditional MapReduce, which is the default-computing engine for Hadoop framework. In contrast to MapReduce, the core data abstraction of Apache Spark, which is a distributed data frame, goes far beyond batch applications to support a variety of compute-intensive tasks, including interactive queries, streaming, machine learning, and graph processing.

Filtered by the chosen hashtag, the public streaming API of Spark collected our data. In order to process the input data into parallel, five computers are used.

Table 1: The processing stages of the proposed methodology

Stage	Definition
Tokenization	The text of each of the tweets is converted into a list of tokens.
Stop Words Removal	In an effort to decrease the dimensionality of the problem, "stop words", i.e. the words so common that their presence does not tell us anything about the dataset are removed.
Stemming	The process of reducing derived words to their stem or root. Stemming is performed after stop words removal.
Filtering	In the filtering phase, the unnecessary characters are filtered out.
Conversion into TDM	The filtered stream is converted into a structured called TDM (Term Document Matrix). TDM represents the terms and frequency of each word in the corpus, in a structured way.
Pattern Analysis	Pattern analysis is conducted to find the most important keywords using the TF-IDF statistics

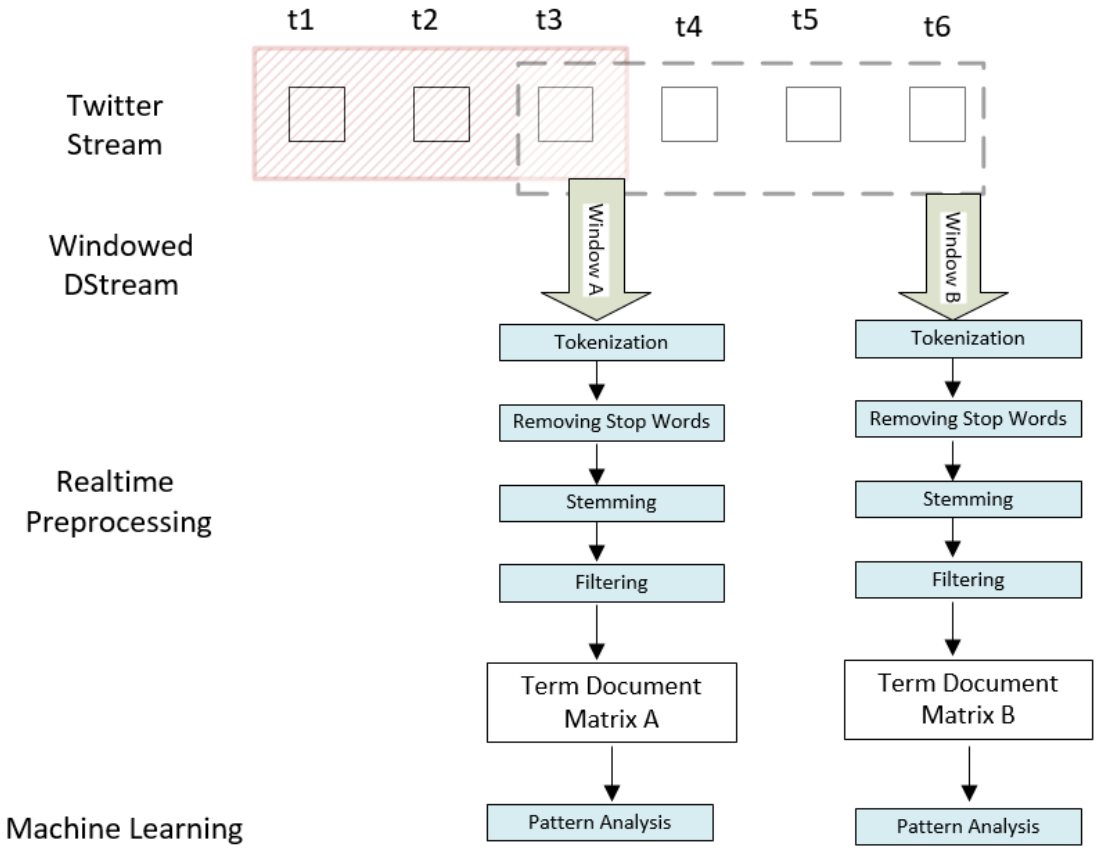

Figure 1 The StreamSensing Approach

The input stream is converted into Resilient Distributed Datasets(RDD), which is the core Spark abstraction. RDD is an immutable fault tolerant collection of elements that can be operated in parallel. It is split into partitions and distributed across the five nodes. They reside in memory and therefore can be processed very fast. The processing operations on RDDs are automatically parallelized across the five nodes by the Spark framework. Before Spark starts processing, it creates an optimized execution plan. It generates graph of RDDs to represent the computation in the form of a Direct Acyclic Graph (DAG). Based on this generated DAG, the required processing and pattern analysis is divided into various tasks, which are scheduled and executed by the Spark framework. The task is to estimate the volume of tweets for a given hashtag in the last five minutes and identify the most popular five keywords for that hashtag specified time-interval. The velocity of incoming data depends on the popularity of the chosen hashtag.

Table 2: Most Important keywords in three different hashtags based on TF-IDFs

#BigData

Keyword	TF-IDF
Computing	0.025
Emerge	0.032
AI	0.034
Machine Learning	0.045
IBM	0.0023

#Ottawa

Keyword	TF-IDF
LRT	0.031
CHEO	0.033
Trudeau	0.041
Senators	0.055
Flood	0.061

#NewYork

Keyword	TF-IDF
Accident	0.037
Traffic	0.023
Manhattan	0.046
Broadway	0.043
Trump	0.065

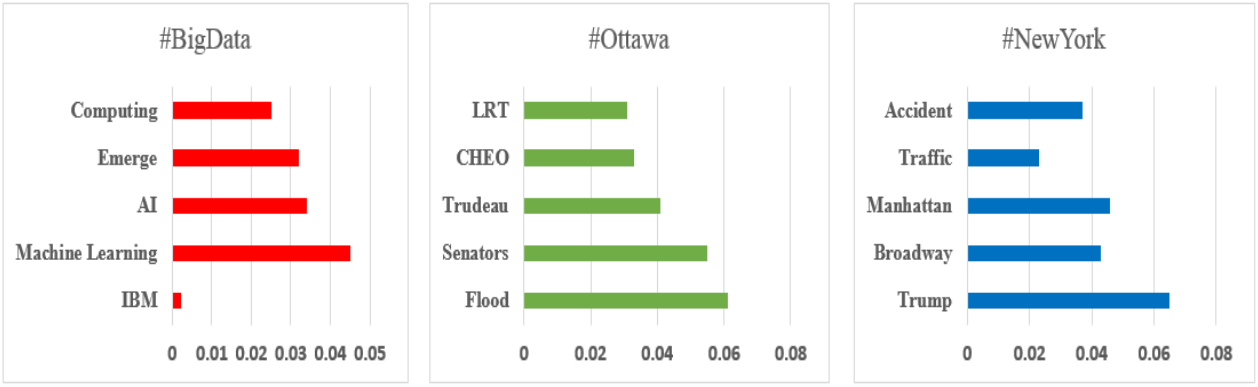

Figure 2: Most Important keywords in three different hashtags based on TF-IDFs

6. ANALYSIS AND RESULTS

Given a chosen hashtag, the importance of a keyword is given by its presence within the tweet and the collection of all tweets being analyzed. Wu et al. (2008) states that a keyword is a good candidate to represent a document if it shows a high frequency within the document, but is rare across the collection. Therefore, two scores, TF and IDF, are calculated for each keyword during the conduct of the two experiments. The results of the three chosen hashtags, #BigData, #NewYork and #Ottawa are shown in Figure 2 and Table 2.

During the first experiment conduct, 3266 tweets were captured for the hashtag #NewYork (high popularity), while 186 tweets were captured for the hashtag #BigData (low popularity). In the second experiment, 1233 tweets were captured for the hashtag #Ottawa (medium popularity). Picking different levels of popularity (high, medium and low) serves the purpose of testing the implementability of StreamSensing approach for scaling and handling different levels of popularity. Using public API of twitter, the streams associated with these hashtags were collected and updated after each minute. After updating and processing this data, pattern analysis was conducted for the data collected in the last five minutes. Therefore, five minutes is the size of the sliding window, which is updated after each minute.

Once the data was captured, the following steps were executed:

1) A TDM was updated after each minute for summarizing the occurrences of words during the last five minutes for a particular hashtag.

2) From the TDM, top 20 words extracted from all the tweets in the last five minutes for the particular hashtag were listed.

3) TF-IDF of each of the words was calculated. Note that if a word has appeared in more than one tweets, it will have a TF-IDF for each of the occurrence. However, the max of the TF-IDF scores is chosen. Then, the words ranked in the first five positions based on their TF-IDF scores were selected and linked to their hashtag.

The results, illustrated in Figure 2 and Table 2, show that for hashtag #BigData, the keywords "Machine Learning", "AI", "Emerge", "Computing", and "IBM" were the top five words that would summarize the tweets regarding #BigData at that instance of time. For the hashtag #NewYork, the keywords "Trump", "Manhattan", "Broadway", "Accident", and "Traffic", were the top five words that would summarize the tweets in that instance of time. Finally, the top five words summarizing the tweets related to the hashtag #Ottawa were "Flood", "Senators", "Trudeau", "CHEO", and "LRT" during the conduct of the experiment.

These results are dynamic and time-related. They can be used to summarize the trends coming from tweets of various hashtags of tweets in real-time. This tool of linking a hashtag with its most important keywords can make communication filterable, organized, and more understandable. It can also be put to great use for getting feedback and suggestions in real-time.

7. CONCLUSION

This paper contributes to the literature from two perspectives theoretically and practically.

From theoretic perspective, we introduce to the literature the approach we call StreamSensing. This approach is a multi-stage mechanism, synthesized from prior research and proposed by authors, that is deemed to be appropriate for analyzing and discovering real-time noisy streams. StreamSensing consists of six stages: (1) tokenization, (2) stop words removal, (3) stemming, (4) filtering, (5) conversion into Term Document Matrix (TDM), and finally (6) pattern analysis. The approach was experimentally tested using real-time Twitter stream data via the fast in-memory processing system called Spark.

From practical perspective, this approach can be used to perform trend analysis on any real-time text data stream. The proposed architecture is flexible and its compute dimension is capable to process distributed in-memory data structures created on-the-fly by the streaming data. It means that depending upon the processing requirements, the degree of parallelism can be adjusted by increasing or decreasing nodes. It can be used for more sophisticated machine learning algorithms with higher processing requirements and can be extended to process multimedia streaming data. It can be extended, as well, to conduct sentiment analyses. For example, instead of capturing the keywords, the sentiments carried by each of the tweet can be analyzed and recorded, which can summarize the feeling about specific hashtags in real-time.

8. REFERENCES

[1] Althoff, T., Borth, D., Hees, J., Dengel, A., (2013). **Analysis and Forecasting of Trending Topics in Online Media Streams**. In MM '13 Proceedings of the 21st ACM international conference on Multimedia. Pages 907-916.

[2] Atefeh, F., Khreich, W., (2013). **A Survey of Techniques for Event Detection in Twitter**. In Journal Computational Intelligence, Volume 31 Issue 1, Pages 132-164.

[3] Becker, H., M. Naaman, and L. Gravano. (2011a). **Beyond trending topics: Real-world event identification on Twitter**. In ICWSM, Barcelona, Spain.

[4] Becker, H., Naaman, M., Gravano, L, (2011). **Beyond Trending Topics: Real-World Event Identification on Twitter**, In Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media. Pages 438-441.

[5] Bollen, J., Mao, H., (2010). **Twitter mood as a stock market predictor**. IEEE Computer, 44(10). Pages 91–94.

[6] Castillo, C., M. Mendoza, and B. Poblete. (2011). **Information credibility on Twitter**. In Proceedings of the 20th International Conference on World Wide Web, WWW '11, ACM, New York, NY, pp. 675–684.

[7] Chen, J., Nairan, R., Nelson, L., Bernstein, M., Chi, E., (2010). **Short and tweet: experiments on recommending content from information streams**. In CHI '10 Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. Pages 1185-1194.

[8] Cordeiro, M. (2012). **Twitter event detection: Combining wavelet analysis and topic inference summarization**. In Doctoral Symposium on Informatics Engineering, DSIE'2012.

[9] Farzindar, A. (2012). **Industrial perspectives on social networks**. In EACL 2012 - Workshop on Semantic Analysis in Social Media.

[10] Fire, M., Goldschmidt, M., Elovici, Y, (2014). **Online Social Networks: Threats and Solutions**. IEEE Communications Surveys & Tutorials (Volume: 16, Issue: 4, Fourthquarter 2014).

[11] H. Wu and R. Luk and K. Wong and K. Kwok. **Interpreting TF-IDF term weights as making relevance decisions**. ACM Transactions on Information Systems, 26 (3). 2008.

[12] Hurlock, J., and M. Wilson. (2011). **Searching Twitter: separating the tweet from the chaff**. In International AAAI Conference on Weblogs and Social Media, Barcelona, Spain.

[13] Johnson, S. (2009). **How Twitter will change the way we live**. http://www.time.com/time/magazine/article /0,9171,1902818,00.html.

[14] Kwak, H., Lee, C., Park, H., and Moon, S. (2010). **What is twitter, a social network or a news media?** In Proceedings of the 19th International Conference on the World Wide Web. 591–600.

[15] Lee, K., B. Eoff, and J. Caverlee. (2011). **Seven months with the devils: A long-term study of content polluters on Twitter**. In International AAAI Conference on Weblogs and Social Media, Barcelona, Spain.

[16] Li, H.-F. and Lee, S.-Y. (2009). **Mining frequent itemsets over data streams using efficient window sliding techniques**. Expert Syst. Appl. 36, 2, 1466–1477.

[17] Long, R., H. Wang, Y. Chen, O. Jin, and Y. Yu. (2011). **Towards effective event detection, tracking and summarization on microblog data**. In Web-Age Information Management, Vol. 6897 of Lecture Notes in Computer Science. Edited by WANG, H., S. LI, S. OYAMA, X. HU, and T. QIAN. Springer: Berlin/Heidelberg, pp. 652–663.

[18] Mittal, A., Goel, A. (2012). **Stock Prediction Using Twitter Sentiment Analysis**. Working Paper Stanford University CS 229.

[19] Muthukrishnan, S., (2005), **Data Streams: Algorithms and Applications**. Now Publ., Boston, MA.

[20] Popescu, A.-M. and Pennacchiott, M. (2011). **Dancing with the stars, NBA games, politics: An exploration of Twitter users' response to events**. In Proceedings of the 5th International AAAI Conference on Weblogs and Social Media. 594–597.

[21] Reddy, V., Rao, T., Govardhan, A. (2014). **Mining Frequent Itemsets (MFI) Over Data Streams: Variable Window Size (VWS) By Context Variation Analysis (CVA) Of The Streaming Transactions.** In International Journal of Data Mining & Knowledge Management Process (IJDKP) Vol.4, No.4. Pages 17-25.

[22] Schubert, E., Schubert, M., Kriegel, H., (2014). **SigniTrend: Scalable Detection of Emerging Topics in Textual Streams by Hashed Significance.** Thresholds In KDD '14 Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining. Pages 871-880.

[23] Zhu, Y. and Shasha, D. (2002). **Statstream: Statistical monitoring of thousands of data streams in real time**. In Proceedings of the 28th Very Large Data Base Conference. 358–369.