Scientific Research Publishing

Journal of Data Analysis and Information Processing, 2018, 6, 46-66
http://www.scirp.org/journal/jdaip
ISSN Online: 2327-7203
ISSN Print: 2327-7211

Integrated Real-Time Big Data Stream Sentiment Analysis Service

Sun Sunnie Chung, Danielle Aring

Department of Electrical Engineering and Computer Science, Cleveland State University, Cleveland, USA
Email: tanxingang@163.com

How to cite this paper: Chung, S.S. and Aring, D. (2018) Integrated Real-Time Big Data Stream Sentiment Analysis Service. *Journal of Data Analysis and Information Processing*, **6**, 46-66.
https://doi.org/10.4236/jdaip.2018.62004

Received: April 20, 2018
Accepted: May 28, 2018
Published: May 31, 2018

Abstract

Opinion (sentiment) analysis on big data streams from the constantly generated text streams on social media networks to hundreds of millions of online consumer reviews provides many organizations in every field with opportunities to discover valuable intelligence from the massive user generated text streams. However, the traditional content analysis frameworks are inefficient to handle the unprecedentedly big volume of unstructured text streams and the complexity of text analysis tasks for the real time opinion analysis on the big data streams. In this paper, we propose a parallel real time sentiment analysis system: Social Media Data Stream Sentiment Analysis Service (SMDSSAS) that performs multiple phases of sentiment analysis of social media text streams effectively in real time with two fully analytic opinion mining models to combat the scale of text data streams and the complexity of sentiment analysis processing on unstructured text streams. We propose two aspect based opinion mining models: Deterministic and Probabilistic sentiment models for a real time sentiment analysis on the user given topic related data streams. Experiments on the social media Twitter stream traffic captured during the pre-election weeks of the 2016 Presidential election for real-time analysis of public opinions toward two presidential candidates showed that the proposed system was able to predict correctly Donald Trump as the winner of the 2016 Presidential election. The cross validation results showed that the proposed sentiment models with the real-time streaming components in our proposed framework delivered effectively the analysis of the opinions on two presidential candidates with average 81% accuracy for the Deterministic model and 80% for the Probabilistic model, which are 1% - 22% improvements from the results of the existing literature.

Keywords

Sentiment Analysis, Real-Time Text Analysis, Opinion Analysis, Big Data Analytics

1. Introduction

In the era of the web based social media, user-generated contents in "any" form of user created content including: blogs, wikis, forums, posts, chats, tweets, or podcasts have become the norm of media to express people's opinion. The amounts of data generated by individuals, businesses, government, and research agents have undergone exponential growth. Social networking giants such as Facebook and Twitter had 1.86 and 0.7 billion active users as of Feb. 2018. The user-generated texts are valuable resources to discover useful intelligence to help people in any field to make critical decisions. Twitter has become an important platform of user generated text streams where people express their opinions and views on new events, new products or news. Such new events or news from announcing political parties and candidates for elections to a popular new product release are often followed almost instantly by a burst in Twitter volume, providing a unique opportunity to measure the relationship between expressed public sentiment and the new events or the new products.

Sentiment analysis can help explore how these events affect public opinion or how public opinion affects future sales of these new products. While traditional content analysis takes days or weeks to complete, opinion analysis of such streaming of large amounts of user-generated text have commanded research and development of a new generation of analytics methods and tools to process them in real-time or near-real time effectively.

Big data is often defined with the three characteristics: volume, velocity and variety [1] [2] because of the nature of being constantly generated massive data sets having large, varied and complex structures or often unstructured (e.g. tweet text). Those three characteristics of big data imply difficulties of storing, analyzing and visualizing for further processes and results with traditional data analysis systems. Common problems of big data analytics are firstly, traditional data analysis systems are not reliable to handle the volume of data to process in an acceptable rate. Secondly, big data processing commonly requires complex data processing in multi phases of data cleaning, preprocessing, and transformation since data is available in many different formats either in semi-structured or unstructured. Lastly, big data is constantly generated at high speed by systems giving that none of the traditional data preprocessing architectures are suitable to efficiently process in real time or near real time.

Two common approaches to process big data are batch-mode big data analytics and streaming-based big data analytics. Batch processing is an efficient way to process high volumes of data where a group of transactions is collected over time [3]. Frameworks that are based on a parallel and distributed system architecture such as Apache Hadoop with MapReduce currently dominate batch mode big data analytics. This type of big data processing addresses the volume and variety components of big data analytics but not velocity. In contrast, stream processing is a model that computes a small window of recent data at one time [3]. This makes computation real time or near-real time. In order to meet the demands of the real-time constraints, the stream-processing model must be able

0to calculate statistical analytics on the fly, since streaming data like user generated content in the form of repeated online user interactions is continuously arriving at high speed [3].

This notable "high velocity" on arrival characteristic of the big data stream means that corresponding big data analytics should be able to process the stream in a single pass under strict constraints of time and space. Most of the existing works that leverage the distributed parallel systems to analyze big social media data in real-time or near real-time perform mostly statistical analysis in real time with pre-computed data warehouse aggregations [4] [5] or simple frequency based sentiment analysis model [6]. More sophisticated sentiment analyses on the streaming data are mostly the MapReduce based batch mode analytics. While it is common to find batch mode data processing works for the sophisticated sentiment analysis on social media data, there are only a few works that propose the systems that perform complex real time sentiment analysis on big data streams [7] [8] [9] and little work is found in that the proposed such systems are implemented and tested with real time data streams.

Sentiment Analysis otherwise known as opinion mining commonly refers to the use of natural language processing (NLP) and text analysis techniques to extract, and quantify subjective information in a text span [10]. NLP is a critical component in extracting useful viewpoints from streaming data [10]. Supervised classifiers are then utilized to predict from labeled training sets. The polarity (positive or negative opinion) of a sentence is measured with scoring algorithms to measure a polarity level of the opinion in a sentence. The most established NLP method to capture the essential meaning of a document is the bag of words (or bag of n-gram) representations [11]. Latent Dirichlet Allocation (LDA) [12] is another widely adopted representation. However, both representations have limitations to capture the semantic relatedness (context) between words in a sentence and suffer from the problems such as polysemy and synonymy [13].

A recent paradigm in NLP, unsupervised text embedding methods, such as Skip-gram [14] [15] and Paragraph Vector [16] [17] to use a distributed representation for words [14] [15] and documents [16] [17] are shown to be effective and scalable to capture the semantic and syntactic relationships, such as polysemy and synonymy, between words and documents. The essential idea of these approaches comes from the distributional hypothesis that a word is represented by its neighboring (context) words in that you shall know a word by the company it keeps [18]. Le and Mikolov [16] [17] show that their method, Paragraph Vectors, can be used in classifying movie reviews or clustering web pages. We employed the pre-trained network with the paragraph vector model [19] for our system for preprocessing to identify n-grams and synonymy in our data sets.

An advanced sentiment analysis beyond polarity is the aspect based opinion mining that looks at other factors (aspects) to determine sentiment polarity such as "feelings of happiness sadness, or anger". An example of the aspect oriented opinion mining is classifying movie reviews based on a thumbs up or downs as seen in the 2004 paper and many other papers by Pang and Lee [10] [20].

Another technique is the lexical approach to opinion mining developed famously by Taboda *et al.* in their SO-CAL calculator [21]. The system calculated semantic orientation, *i.e.* subjectivity, of a word in the text by capturing the strength and potency to which a word was oriented either positively or negatively towards a given topic, using advanced techniques like amplifiers and polarity shift calculations.

The single most important information needs to be identified in a sentiment analysis is to find out about opinions and perspectives on a particular topic otherwise known as topic-based opinion mining [22]. Topic-based opinion mining seeks to extract personal viewpoints and emotions surrounding social or political events by semantically orienting user-generated content that has been correlated by topic word(s) [22].

Despite the success of these sophisticated sentiment analysis methods, little is known about whether they may be scalable to apply in the multi-phased opinion analysis process to a huge text stream of user generated expressions in real time. In this paper, we examined whether a stream-processing big data social media sentiment analysis service can offer scalability in processing these multi-phased top of the art sentiment analysis methods, while offering efficient near-real time data processing of enormous data volume. This paper also explores the methodologies of opinion analysis of social network data. To summarize, we make the following contributions:

- We propose a fully integrated, real time text analysis framework that performs complex multi-phase sentiment analysis on massive text streams: Social Media Data Stream Sentiment Analysis Service (SMDSSAS).
- We propose two sentiment models that are combined models of topic, lexicon and aspect based sentiment analysis that can be applied to a real-time big data stream in cooperation with the most recent natural language processing (NLP) techniques:
  - Deterministic Topic Model that accurately measures user sentiments in the subjectivity and the context of user provided topic word(s).
  - Probabilistic Topic Model that effectively identifies polarity of sentiments per topic correlated messages over the entire data streams.
- We fully experimented on the popular social media Twitter message streams captured during the pre-election weeks of the 2016 Presidential Election to test the accuracy of our two proposed sentiment models and the performance of our proposed system SMDSSAS for the real time sentiment analysis. The results show that our framework can be a good alternative for an efficient and scalable tool to extract, transform, score and analyze opinions for the user generated big social media text streams in real time.

2. Related Works

Many existing works in the related literature concentrate on topic-based opining mining models. In topic-based opinion mining, sentiment is estimated from the

messages related to a chosen topic of interest such that topic and sentiment are jointly inferred [22]. There are many works on the topic based sentiment analysis where the models are tested on a batch method as listed in the reference Section. While there are many works in the topic based models for batch processing systems, there are few works in the literature on topic-based models for real time sentiment analysis on streaming data. Real-time topic sentiment analysis is imperative to meet the strict time and space constraints to efficiently process streaming data [6]. Wang *et al.* in the paper [6] developed a system for Real-Time Twitter Sentiment Analysis of the 2012 Presidential Election Cycle using the Twitter firehose with a statistical sentiment model and a Naive Bayes classifier on unigram features. A full suite of analytics were developed for monitoring the shift in sentiment utilizing expert curated rules and keywords in order to gain an accurate picture of the online political landscape in real time. However, these works in the existing literature lacked the complexity of sentiment analysis processes. Their sentiment analysis model for their system is based on simple aggregations for statistical summary with a minimum primitive language preprocessing technique.

More recent research [23] [24] have proposed big data stream processing architectures. The first work in 2015 [23] proposed a multi-layered storm based approach for the application of sentiment analysis on big data streams in real time and the second work in 2016 [24] proposed a big data analytics framework (ASMF) to analyze consumer sentiments embedded in hundreds of millions of online product reviews. Both approaches leverage probabilistic language models by either mimicking "document relevance": with probability of the document generating a user provided query term found within the sentiment lexicon [23] or by adapting a classical language modeling framework to enhance the prediction of consumer sentiments [24]. However, the major limitation of their works is both the proposed frameworks have never been implemented and tested under an empirical setting or in real time.

3. Architecture of Big Data Stream Analytics Framework

In this Section, we describe the architecture of our proposed big data analytics framework that is illustrated in **Figure 1**. Our sentiment analysis service, namely Social Media Data Stream Sentiment Analysis Service (SMDSSAS) consists of six layers—Data Storage/Extraction Layer, Data Stream Layer, Data Preprocessing and Transformation Layer, Feature Extraction Layer, Prediction Layer, and Presentation Layer. For these layers, we employed well-proven methods and tools for real time parallel distributed data processing.

For the real time data analytics component, SMDSSAS leverages the Apache Spark [1] [7] and a NoSQL Hive [25] big data ecosystem, which allows us to develop a streamlined pipelining with the natural language processing techniques for fully integrated complex multiphase sentiment analysis that store, process and analyze user generated content from the Twitter streaming API.

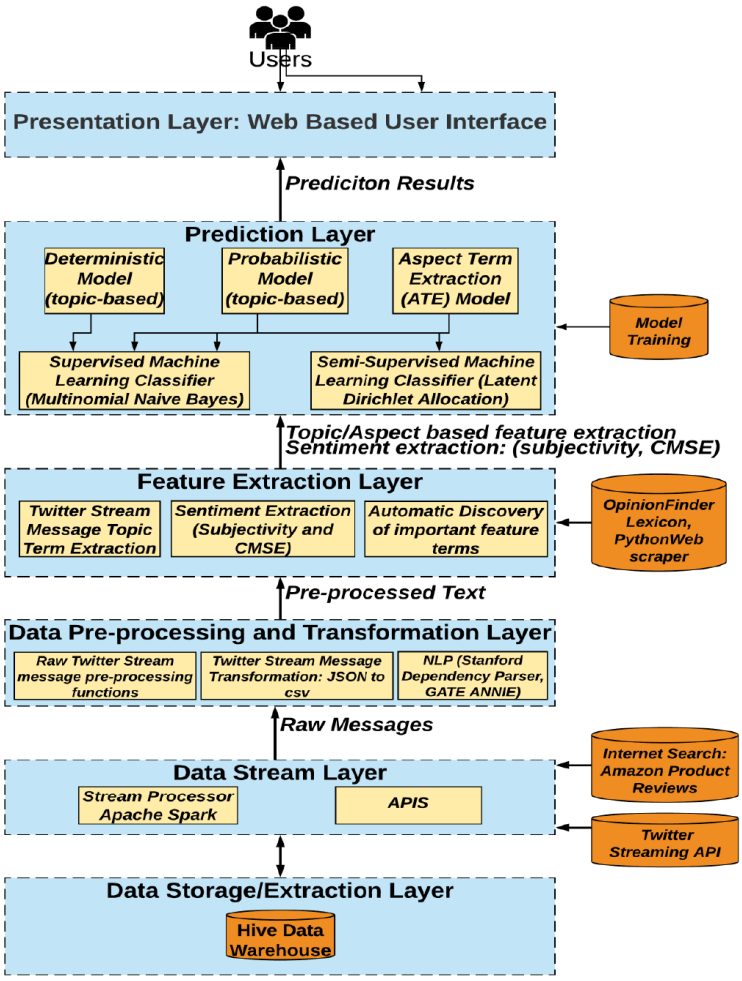


Figure 1. Architecture of social media data stream sentiment analysis service (SMDSSAS).

The first layer is Data Storage/Extraction Layer for extraction of user tweet fields from the Twitter Stream that are to be converted to topic filtered DStreams through Spark in the next Data Stream layer. DStream is a memory unit of data in Spark. It is the basic abstraction in Spark Streaming, which is a continuous sequence of Resilient Distributed datasets (RDDs of the same type) that represents a continuous stream of data. The extracted Tweet messages are archived in Hive's data warehouse store via Cloudera's interactive web based analytics tool Hue and direct streaming into HDFS.

The second Layer: Data Stream Layer processes the extracted live streaming of user-generated raw text of Twitter messages to Spark contexts and DStreams. This layer is bidirectional with both the Data Storage/Extraction layer and the next layer the Data Preprocessing and Transformation Layer.

The third layer: Data Preprocessing and Transformation layer is in charge of building relationships in the English natural language and cleaning raw text twitter messages with the functions to remove both control characters sensitive to Hive data warehouse scanner and non-alphanumeric characters from. We

employee the natural language processing techniques in the Data Preprocessing layer with the pertained network in the paragraph vector model [16] [17]. This layer can also employee the Stanford Dependency Parser [26] and Named Entity Recognizer [27] to build an additional pipelining of Dependency, Tokenizer, Sentence Splitting, POS tagging and Semantic tagging to build more sophisticated syntax relationships in the Data Preprocessing stage. The transformation component of this later preprocesses in real time the streaming text in JSON to CSV formatted Twitter statuses for Hive table inserts with Hive DDL. The layer is also in charge of removing ambiguity of a word that is determined with pre-defined word corpuses for the sentiment scoring process later.

The forth Layer, Feature Extraction layer, is comprised of a topic based feature extraction function for our Deterministic and Probabilistic sentiment models. The topic based feature extraction method employees the Opinion Finder Subjectivity Lexicon [28] for identification and extraction of sentiment based on the related topics of the user twitter messages.

The fifth layer of our framework: the Prediction Layer uses our two topic and lexicon based sentiment models: Deterministic, and Probabilistic for sentiment analysis. The accuracy of each model was measured using the supervised classifier Multinomial Naive Bayes to test the capability of each model for correctly identifying and correlating users' sentiments on the topics related data streams with a given topic (event).

Our sixth and final layer is Presentation layer that consists of a web based user interface.

4. Sentiment Model

Extracting useful viewpoints (aspects) in context and subjectivity from streaming data is a critical task for sentiment analysis. Classical approaches of sentimental analysis have their own limitations in identifying accurate contexts, for instance, for the lexicon-based methods; common sentiment lexicons may not be able to detect the context-sensitive nature of opinion expressions. For example, while the term "small" may have a negative polarity in a mobile phone review that refers to a "small" screen size, the same term could have a positive polarity such as "a small and handy notebook" in consumer reviews about computers. In fact, the token "small" is defined as a negative opinion word in the well-known sentiment lexicon list Opinion-Finder [28].

The sentiment models developed for SMDSSAS are based on the aspect model [29]. The aspect based opinion mining techniques are to identify to extract personal opinions and emotions of surrounding social or political events by capturing semantically orienting contents in subjectivity and context that are correlated by aspects, *i.e.* topic words. The design of our sentiment model was based on the assumption that positive and negative opinions could be estimated per a context of a given topic [22]. Therefore, in generating data for model training and testing, we employed a topic-based approach to perform sentiment annota-

tion and quantification on related user tweets.

The aspect model is a core of probabilistic latent semantic analysis in the probabilistic language model for general co-occurrence data which associates a class (topic) variable $t \in T = \{t_1, t_2, \cdots, t_k\}$ with each occurrence of a word $w \in W = \{w_1, w_2, \cdots, w_m\}$ in a document $d \in D = \{d_1, d_2, \cdots, d_n\}$. The Aspect model is a joint probability model that can be defined in selecting a document d with probability $P(d)$, picking a latent class (topic) t with probability $P(t|d)$, and occurring a word (token) w with probability $P(w|t)$.

As a result one obtains an observed pair (d,w), while the latent class variable z is discarded. Translating this process into a joint probability model results in the expression as follow:

$$P(d,w) = P(d)P(w|d) \tag{1}$$

where

$$P(w|d) = \Sigma_{t\in T} P(w|t)P(t|d) \tag{2}$$

Essentially, to derive (2) one has to sum over the possible choices of z that could have generated the observation.

The aspect model is based on two independence assumptions: First, any pairs (d,w) are assumed to be occurred independently; this essentially corresponds to the bag-of-words (or bag of n-gram) approach. Secondly, the conditional independence assumption is made that conditioned on the latent class t, words w are occurred independently of the specific document identity d. Given that the number of class states is smaller than the number of documents ($K \ll N$), t acts as a bottleneck variable in predicting w conditioned on d.

Following the likelihood principle, $P(d)$, $P(t|d)$, and $P(w|t)$ can be determined by maximization of the log-likelihood function

$$L = \Sigma_{d\in D}\Sigma_{w\in W} n(d,w)\log P(d,w) \tag{3}$$

where $n(d,w)$ denotes the term frequency, *i.e.*, the number of times w occurred in d. An equivalent symmetric version of the model can be obtained by inverting the conditional probability $P(t|d)$ with the Bayes' theorem, which results in

$$P(d,w) = \Sigma_{t\in T} P(t)P(w|t)P(d|t) \tag{4}$$

In the Information Retrieval context, this Aspect model is used to estimate the probability that a document d is related to a query q [2]. Such a probabilistic inference is used to derive a weighted vector in Vector Space Model (VSM) where a document d contains a user given query q [2] where q is a phrase or a sentence that is a set of classes (topic words) as $d \cap q = T = \{t_1, t_2, \cdots, t_k\}$.

$$score(q,d) = \Sigma_{t\in d\cap q} tf.idf_{t,d} \tag{5}$$

where $tf.idf_{t,d}$ is defined as a term weight $w_{t,d}$ of a topic word t with $tf_{t,d}$ being the term frequency of a topic word t_j occurs in d_i and idf_t, being the inverted document index defined with df_t the number of documents that contain t_j as below. N is the total number of documents.

$$w_{t,d} = \log\left(1 + tf_{t,d}\right) * \log_{10}\left(N/df_t\right) \quad (6)$$

Then d and q are represented with the weighted vectors for the common terms. $score(q,d)$ can be derived using the cosine similarity function to capture the concept of document "relevance" of d respect to q in the context of topic words in q. Then the cosine similarity function is defined as the score function with the length normalized weighted vectors of q and d as follow.

$$\cos(\boldsymbol{q},\boldsymbol{d}) = \frac{\boldsymbol{q}\cdot\boldsymbol{d}}{|\boldsymbol{q}||\boldsymbol{d}|} = \frac{\boldsymbol{q}}{|\boldsymbol{q}|}\cdot\frac{\boldsymbol{d}}{|\boldsymbol{d}|} = \frac{\sum_{i=1}^{|V|} q_i d_i}{\sqrt{\sum_{i=1}^{|V|} q_i^2}\sqrt{\sum_{i=1}^{|V|} d_i^2}} \quad (7)$$

4.1. Context Identification

We derive a topic set $T(q)$ by generating a set of all the related topic words from a user given query (topics) $q = \{t_1, t_2, \cdots, t_k\}$ where q is a set of tokens. For each token t_i in q, we derive the related topic words to add to the topic set $T(q)$ based on the related language semantics $R(t_i)$ as follow.

$$R(t_i) := label\_synonym(t_i)|t_i|t_i.*|*.t_i|t_i\_t_j|(t_i|t_j)|t_i*|t_i + |\#.t_i|@.t_i \quad (8)$$

where $t_i, t_j \in T$. $t_i.*|*.t_i$ denote any word concatenated with t_i and $t_i\_t_j$ denotes a bi-gram with t_i and t_j, $label\_synonym(t_i)$ is a set of the labeled synonyms of t_i in the dictionary identified by in WordNet [23]. For context identification, we can choose to employee the pre-trained network with the paragraph vector model [16] [17] for our system for preprocessing. The paragraph vector model is more robust in identifying synonyms of a new word that is not in the dictionary.

4.2. Measure of Subjectivity in Sentiment: *CMSE* and *CSOM*

The design of our experiments of each model were intended to capture social media Twitter data streams of a surrounding special social or political event, so we targeted to capture data streams to test during two special events—the 2016 US Presidential election and the 2017 Inauguration. The real time user tweet streams were collected from Oct. 2016 to Jan. 2017. The time frames chosen are a pre-election time of the October 23rd week and the pre-election week of November 5th, as well as a pre-inauguration time the first week of January 2017.

We define the document-level polarity $sentiment(d_i)$ with a simple polarity function that simply counts the number of positive and negative words in a document (a twitter message) to determine an initial sentiment measure $sentiment(d_i)$ and the sentiment label $sentiment_i$ for each document d_i as follow:

$$sentiment(d_i) = \sum_{k=1}^{m}\left(Pos(w_k) + Neg(w_k)\right) = FreqPos(d_i) - FegNeg(d_i) \quad (9)$$

where d_i is a document (message) in a tweet stream D of a given topic set T with $1 \le i < n$ and $d_i = \{w_1, \cdots, w_m\}$, m is the number of words in d_i. $Pos(w_k) = 1$ if w_k is a positive word and $Neg(w_k) = -1$ if w_k is a negative word. $sentiment(d_i)$ is the difference between the frequency of the positive words denoted as $FreqPos(d_i)$ and the frequency of negative words denoted as $FreqNeg(d_i)$ in d_i indicating an

initial opinion polarity measure with $-m \le sentiment(d_i) \le m$ and a sentiment label of d_i $sentiment_i = 1$ for positive if $sentiment(d_i) \ge 1$, 0 if $sentiment(d_i) = 0$ for neutral, and -1 for negative if $sentiment(d_i) \le -1$.

Then, we define $w(d_i)$ a weight for a sentiment orientation for d_i to measure a subjectivity of sentiment orientation of a document, then a weighted sentiment measure for d_i $senti\_score(d_i)$ is defined with $w(d_i)$ and $sentiment_i$ the sentiment label of d_i as a score of sentiment of d_i as follow:

$$w(d_i) = \alpha * \left(\frac{sentiment(d_i)}{\sqrt{FreqPos(d_i)^2 + FreqNeu(d_i)^2 + FreqNeg(d_i)^2}} \right) \tag{10}$$

$$senti\_score(d_i) = sentiment_i + w(d_i) \tag{11}$$

where $-1 \le w(d_i) \le 1$, and α is a control parameter for learning. When $\alpha = 0$, $senti\_score(d_i) = sentiment_i$, $senti\_score(d_i)$ gives more weight toward a short message with strong sentiment orientation. $w(d_i) = 0$ for neural.

Class Max Sentiment Extraction (*CMSE*): To test the performance of our models and to predict the outcomes of events such as the 2016 Presidential election from the extracted user opinions embedded in tweet streams, we quantify the level of the sentiment in the data set with Class Max Sentiment Extraction (*CMSE*) to generate statistically relevant absolute sentiment values to measure an overall sentiment orientation of a data set for a given topic set for each sentiment polarity class to compare among different data sets. To quantify a sentiment of a data set D of a given topic set T, we define $CMSE(D(T))$ as follow.

For a given Topic set T, for each $d_i \in D(T)$ where d_i contains at least one of the topic words of interest in T in a given tweet stream D, $CMSE(D(T))$ returns a weighted sum of $senti\_score(d_i)$ of the data set D on T as follow:

$$CMSEpos(D(T)) = \sum_{i=1}^{n} (senti\_score(d_i)) * \delta(sentiment_i = 1)$$
$$\text{where } \delta = 1 \text{ if } sentiment_i = pos, \text{ otherwise } 0 \tag{12}$$

$$CMSEneg(D(T)) = \sum_{i=1}^{n} (senti\_score(d_i)) * \delta(sentiment_i = -1)$$
$$\text{where } \delta = 1 \text{ if } sentiment_i = neg, \text{ otherwise } 0 \tag{13}$$

$$CMSEneu(D(T)) = \sum_{i=1}^{n} (senti\_score(d_i)) * \delta(sentiment_i = 0)$$
$$\text{where } \delta = 1 \text{ if } sentiment_i = pos, \text{ otherwise } 0 \tag{14}$$

where $1 \le i < n$ and $D(T) = \{d_1, \cdots, d_n\}$, n is the number of documents in $D(T)$. *CMSE* is to measure the maximum sentiment orientation values of each polarity class for a given topic correlated data set $D(T)$. It is a sum of the weighted document sentiment scores for each sentiment class—positively labeled d_i, negatively labeled d_i, and neutrally labeled d_i respectively in a given data set $D(T)$ for a user given topic word set T. *CMSE* is the same as an aggregated count of $sentiment_i$ when $\alpha = 0$.

CMSE is an indicator of how strongly positive or negative the sentiment is in a data set for a given topic word set T where $D(T)$ is a set of documents (messages) in a tweet stream where each document $d_i \in D(T)$ $1 \le i \le n$, contains at least

one of the topic words $t_j \in T = \{t_1, \cdots, t_k\}$ with $1 \le j \le k$ and T is a set of all the related topic words derived from a user given query q as a seed to generate T. T_j, which is a subset of T, is a set of topic words that is derived from a given topic $t_j \in T$. $D(T_j)$, a subset of $D(T)$, is a set of documents where each document d_i contains at least one of the related topic words in a topic set T_j. Every topic word set is derived by the Context Identifier described in Section 4.1. With the Donald Trump and Hillary Clinton example, three topic-correlated data sets are denoted as below.

$D(T_j)$ is a set of documents with a topic word set T_j derived from {Donald Trump|Hillary Clinton}.

$D(TR_j)$ is a set of documents, a subset of $D(T_j)$, with a topic word set TR_j derived from {Donald Trump}.

$D(HC_j)$ is a set of documents, a subset of $D(T_j)$, with a topic word set HC_j derived from{Hillary Clinton}.

where m, are the number of document d_i in $D(TR_j)$ and $D(HC_j)$ respectively. For example, $CMSEpos(D(TR_j))$ is the maximum positive opinion measure in the tweet set that are talking about the candidate Donald Trump.

***CSOM* (Class Sentiment Orientation Measure):** *CSOM* is to measure a relative ratio of the level of the positive and negative sentiment orientation for a given topic correlated data set over the entire dataset of interest.

For *CSOM*, we define two relative opinion measures: Semantic Orientation (*SMO*) and Sentiment Orientation (*STO*) to quantify a polarity for a given data set correlated with a topic set T_j. *SMO* indicates a relative polarity ratio between two polarity classes within a given topic data set. *STO* indicates a ratio of the polarity of a given topic set over an entire data set.

With our Trump and Hillary example from the 2016 Presidential Election event, the positive *SMO* for the data set $D(TR_j)$ with the topic word "Donald Trump" and the negative *SMO* for the Hillary Clinton topic set $D(HC_j)$ can be derived for each polarity class respectively as below. For example, the positive *SMO* for a topic set $D(TR_j)$ for Donald Trump and the negative *SMO* for a topic set $D(HC_j)$ for Hillary Clinton are defined as follow:

$$PosSMO\left(TR_j\right) := \frac{CMSE_{pos}\left(D\left(TR_j\right)\right)}{CMSE_{pos}\left(D\left(TR_j\right)\right)+CMSE_{neg}\left(D\left(TR_j\right)\right)+CMSE_{neu}\left(D\left(TR_j\right)\right)} \quad (15)$$

$$NegSMO\left(HC_j\right) := \frac{CMSE_{neg}\left(D\left(HC_j\right)\right)}{CMSE_{pos}\left(D\left(HC_j\right)\right)+CMSE_{neg}\left(D\left(HC_j\right)\right)+CMSE_{neu}\left(D\left(HC_j\right)\right)} \quad (16)$$

When $\alpha = 0$, $senti\_score(d_i) = sentiment_i$, so *CMSE* and *SMO* are generated with count of $sentiment_i$ of the data set. Then, Sentiment Orientation (*STO*) for a topic set $D(TR_j)$ for Donald Trump and the negative *STO* for a topic set $D(HC_j)$ for Hillary Clinton are defined as follow:

$$PosSTO(TR_j) = PosSMO(TR_j) * Weight(TR_j) \quad (17)$$

$$NegSTO(HC_j) = NegSMO(HC_j) * Weight(HC_j) \quad (18)$$

where $Weight(TR_j)$ and $Weight(HC_j)$ are the weights of the topics over the entire dataset, defined as follow. Therefore, $STO(TR_j)$ indicates a weighted polarity of the topic TR_j over the entire data set $D(T_j)$ where $D(T_j) = D(TR_j) \cup D(HC_j)$.

$$Weight(TR_j) := \frac{CMSE_{pos}(D(TR_j)) + CMSE_{neu}(D(TR_j)) + CMSE_{neg}(D(TR_j))}{CMSE_{pos}(D(T_j)) + CMSE_{neg}(D(T_j)) + CMSE_{neu}(D(T_j))} \quad (19)$$

$$Weight(HC_j) := \frac{CMSE_{pos}(D(HC_j)) + CMSE_{neu}(D(HC_j)) + CMSE_{neg}(D(HC_j))}{CMSE_{pos}(D(T_j)) + CMSE_{neg}(D(T_j)) + CMSE_{neu}(D(T_j))} \quad (20)$$

4.3. Deterministic Topic Model

The Deterministic Topic Model considers the context of the words in the texts and the subjectivity of the sentiment of the words given the context. Given the presumption that topic and sentiment can be jointly inferred, the Deterministic Topic Model measures polarity strength of sentiment in the context of user provided topic word(s). Deterministic Topic Model considers subjectivity of each word (token) in d_i in $D(T_j)$. Likelihoods were estimated as relative frequencies with the weighted subjectivity of a word. Using the Opinion Finder [28], lexicon of the tweets were categorized and labeled by subjectivity and polarity. The 6 different weight levels below define subjectivity. Each token was categorized to one of the 6 strength scales and weighted with subjectivity strength scale range from −2 to +2 where −2 denotes "strongest" subjective negative; +2: strongest subjective positive word.

$subjScale(w_t)$ is defined as Subjectivity Strength Scale for each token w_t in d_i. The weight of each group is assigned as below for the 6 subjectivity strength sets. Any token that does not belong to any of the 6 subjectivity strength sets is set to 0.

strSubjPosW:= {set of strong positive subjective words}: +2
wkSubjPosW:= {set of weak positive subjective words}: +1
strSubjNeuW:= {set of strong neutral subjective words}: 0.5
wkSubjNeuW:= {set of weak neutral subjective words}: −0.5
wkSubjNegW:= {set of weak negative subjective words}: −1.
strSubjNegW:= {set of strong negative subjective words}: −2
None = None of above: 0

$$SentimentSubj(d_i) = \sum_{t=1}^{m} subjScale(w_t) \quad (21)$$

m is the number of tokens in d_i. $-m*2 \le SentimentSubj(d_i) \le m*2$. Note that $subjScale(w_t)$ of each neutral word is not 0. We consider a strong neutral

opinion as a weak positive and a weak neutral as a weak negative by assigning very small positive or negative weights. The sentiment of each d_i is then defined by the sum of the frequency of each subjectivity group with its weighted *subjScale*.

$$wSubj\left(d_i\right) = \alpha * \left(\frac{SentimentSubj\left(d_i\right)}{\left\|d_i\right\|}\right) \tag{22}$$

$$senti\_score\_subj\left(d_i\right) = SentimentSubj\left(d_i\right) + wSubj\left(d_i\right) \tag{23}$$

Then $CMSE_{Subj}(D(T))$ is a sum of subjectivity weighted opinion polarity for a given topic set $D(T)$ with $d_i \in D(T)$. It can be defined with $senti\_score\_subj\left(d_i\right)$ as follow.

$$CMSEpos_{subj} = \sum_{i=1}^{n} senti\_score\_subj\left(d_i\right) * \delta\left(sentiment_i = 1\right) \tag{24}$$

where $\delta = 1$ if $sentiment_i = pos$, otherwise 0

$$CMSEneg_{subj} = \sum_{i=1}^{n} senti\_score\_subj\left(d_i\right) * \delta\left(sentiment_i = -1\right) \tag{25}$$

where $\delta = 1$ if $sentiment_i = neg$, otherwise 0

$$CMSEneu_{subj} = \sum_{i=1}^{n} senti\_score\_subj\left(d_i\right) * \delta\left(sentiment_i = 0\right) \tag{26}$$

where $\delta = 1$ if $sentiment_i = neu$, otherwise 0

Then, we define our deterministic model $\rho\varepsilon\left(Pos\_T_j\right)$ as a length normalized sum of subjectivity weighted $senti\_score\_subj\left(d_i\right)$ for a given topic T_j with $d_i \in D\left(T_j\right)$ as follow:

$$\rho\varepsilon\left(Pos\_T_j\right) = \frac{CMSE_{pos_{subj}}\left(D\left(T_j\right)\right)}{CMSE_{pos_{subj}}\left(D\left(T_j\right)\right) + CMSE_{neu_{subj}}\left(D\left(T_j\right)\right) + CMSE_{neg_{subj}}\left(D\left(T_j\right)\right)} \tag{27}$$

where $D(T)$ is a set of documents (messages) in a tweet stream where each document $d_i \in D(T), 1 \le i \le n$, contains one of the topic words $t_j \in T = \{t_1, \cdots, t_k\}$ where $1 \le j \le k$ and T is a set of all the related topic words derived from the user given topics and T_j is a set of all the topic words that are derived from a given query q as defined in the Section 4.1. $D\left(T_j\right)$, a subset of $D(T)$, is a set of documents where each document d_i contains one of the related topic words in a topic set T_j and n is the number of document d_i in $D\left(T_j\right)$.

4.4. Probabilistic Topic Model

The probabilistic model adopts *SMO* and *STO* measures of *CSOM* with the subjectivity to derive a log-based modified log-likelihood of the ratio of subjectivity weighted *PosSMO* and *NegSMO* over a given topic set $D(T)$ and a subset $D(T_j)$.

Our probabilistic model ρ with a given topic set $D(T)$ and $D(T_j)$ measures the probability of sentiment polarity of a given a topic set $D(T_j)$ where $D(T_j)$ is a subset of $D(T)$. For example, the probability of the positive opinion for Trump in $D(T)$, denoted as $P(Pos\_TR)$, is defined as follow:

$$\rho(Pos\_TR) = \frac{CMSEpos_{subj}(D(TR)) + CMSEneu_{subj}(D(TR)) + \epsilon}{CMSEpos_{subj}(D(T)) + CMSEneu_{subj}(D(T)) + CMSEneg_{subj}(D(T))} \quad (28)$$

ϵ is a smoothing factor [30] and we consider a strong neutral subjectivity as a weak positivity here. Then, we define our probabilistic model $\rho(POS\_TR)$ as

$$\rho(POS\_TR) = \log\left(\frac{P(Pos\_TR)}{NegativeInfo(TR)}\right) \quad (29)$$

where NegativeInfo(*TR*) is essentially a subjectivity weighted *NegSMO*(TR_j) defined as follow.

$$NegativeInfo(TR) = \frac{CMSEneg_{subj}(D(TR)) + \epsilon}{CMSEpos_{subj}(D(TR)) + CMSEneu_{subj}(D(TR)) + CMSEneg_{subj}(D(TR))} \quad (30)$$

Our probabilistic model penalizes with the weight of the negative opinion in the correlated topic set *D*(*TR*) when measuring a positive opinion of a topic over a given entire data set *D*(*T*).

$$\rho(Pos\_TR) = \log\left(\frac{\frac{CMSEpos_{subj}(D(TR)) + CMSEneu_{subj}(D(TR)) + \epsilon}{CMSEpos_{subj}(D(T)) + CMSEneu_{subj}(D(T)) + CMSEneg_{subj}(D(T))}}{\frac{CMSEneg_{subj}(D(T)) + \epsilon}{CMSEpos_{subj}(D(TR)) + CMSEneu_{subj}(D(TR)) + CMSEneg_{subj}(D(TR))}}\right) \quad (31)$$

4.5. Multinomial Naive Bayes

The fifth layer of our framework: the Prediction Layer employees the Deterministic and Probabilistic sentiment models discussed in Section 4 to our predictive classifiers for event outcome prediction in a real-time environment. Predictive performance of each model was measured using a supervised predictive analytics model: Multinomial Naive Bayes.

Naive Bayes is a supervised probabilistic learning method popular for text categorization problems in judging if documents belong to one category or another because it is based on the assumption that each word occurrence in a document is independent as in "bag of word" model. Naive Bayes uses a technique to construct a classifier: it assigns class labels to problem instances represented as vectors of feature values where class labels are drawn from a finite set [31]. We utilized the Multinomial model for text classification based on "bag of words" model for a document [32]. Multinomial Naive Bayes models the distribution of words in a document as a multinomial. A document is treated as a sequence of words and it is assumed that each word position is generated independently of every other. For classification, we assume that there are a fixed number of classes, where a class $C_k \in \{C_1, C_2, \cdots, C_m\}$, each with a fixed set of multinomial parameters. The parameter vector for a class $C_k = \{C_{k1}, \cdots, C_{kn}\}$ where *n* is the

size of the vocabulary, and $\sum C_k = 1$.

$$p(x \mid C_k) = \frac{\left(\sum_i x_i\right)!}{\prod_i x_i!} \prod_i pki^{x_i} \tag{32}$$

In a multinomial event model, a document is an ordered sequence of word events, that represent the frequencies which certain events have been generated by a multinomial $(p_1 \cdots p_n)$ where p_i is the probability that event i occurs, and x_i is the feature vector counting the number of times event i was observed in an instance [32]. Each document d_i is drawn from a multinomial distribution of words with as many independent trials as the length of d_i, yielding a "bag of words" representation for the documents [32]. Thus the probability of a document given its class is the representation of k such multinomial [32].

5. Experiments

We applied our sentiment models discussed in Section(s) 4.2 and 4.3 on the real-time twitter stream for the following events—the 2016 US Presidential election and the 2017 Inauguration. User opinion was identified extracted and measured surrounding the political candidates and corresponding election policies in an effort to demonstrate SMDSSAS's accurate critical decision making capabilities.

A total of 74,310 topic-correlated tweets were collected randomly chosen on a continuous 30-second interval in Apache Spark DStream accessing the Twitter Streaming API for the pre-election week of November 2016 and the pre-election month on October, as well as pre-inauguration week in January. The context detector on the following topics generates the set of topic words: Hillary Clinton, Donald Trump and political policies. The number of the topic correlated tweets for the candidate Donald Trump was ~53,009 tweets while the number of the topic correlated tweet for the candidate Hillary Clinton was ~8510 which is a lot smaller than that of Trump.

Tweets were preprocessed with a custom cleaning function to remove all non-English characters including: the Twitter at "@" and hash tag "#" signs, image/website URLS, punctuation: "[. , ! " ']", digits: [0-9], and non-alphanumeric characters: $ % & ^ * () + ~ and stored in NoSql Hive database. Each topic-correlated tweet was labeled for sentiment using the OpinionFinder subjectivity word lexicon and the $subjScale(w_i)$ defined in 4.3 associating a numeric value to each word based on polarity and subjectivity strength.

5.1. Predicting the Outcome of 2016 Presidential Election in Pre-Election Weeks

Figure 2 shows the results of analysis of sentiment orientation on the two presidential candidates for the several months of pre-election 2016 tweet traffic. We noted the lowest positive polarity measure (0.11) for Donald Trump occurred during the pre-election October, but it soared to more than double (0.26) on the election month November and (0.24) on pre-inauguration January 2017. His negative sentiment orientation was already a lot lower (0.022) than his positive

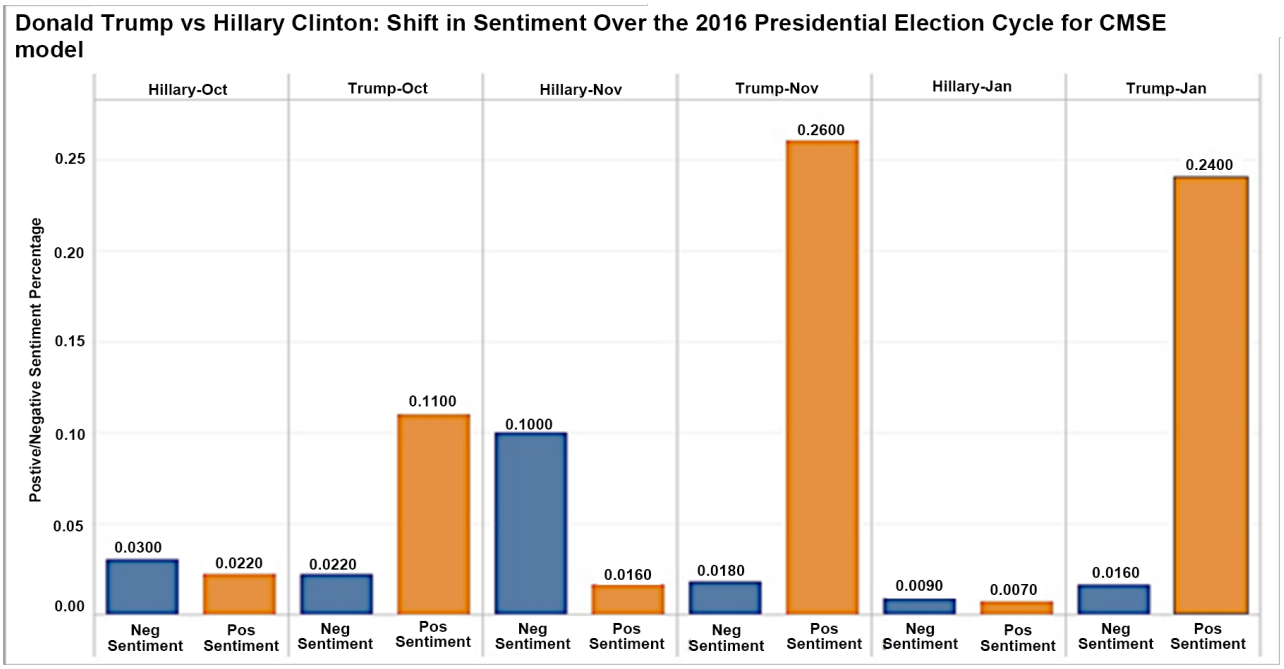


Figure 2. Measuring pre-election sentiment orientation shift for 2016 presidential election cycle.

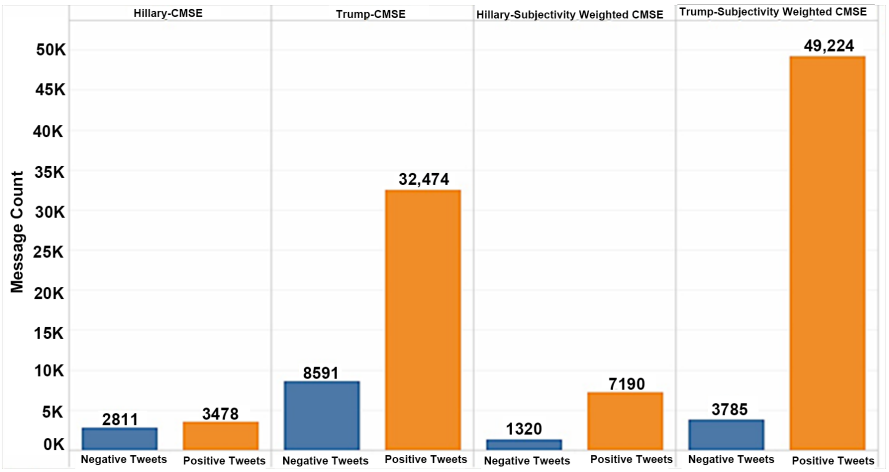
orientation on October and it kept dropping to 0.016 for November and January. In contrast, Hillary Clinton's positive and negative sentiment orientation measures were consistently low during October and November; her positive sentiment measure was ranging from 0.022 on October to 0.016 on November, which is almost ten times smaller than Trump's. It kept dropping to 0.007 on January. Clinton's negative orientation measure was 10 times higher than Trump's ranging from 0.03 on October to 0.01 on November, but it decreased to 0.009 on January.

5.2. Predicting with Deterministic Topic Model

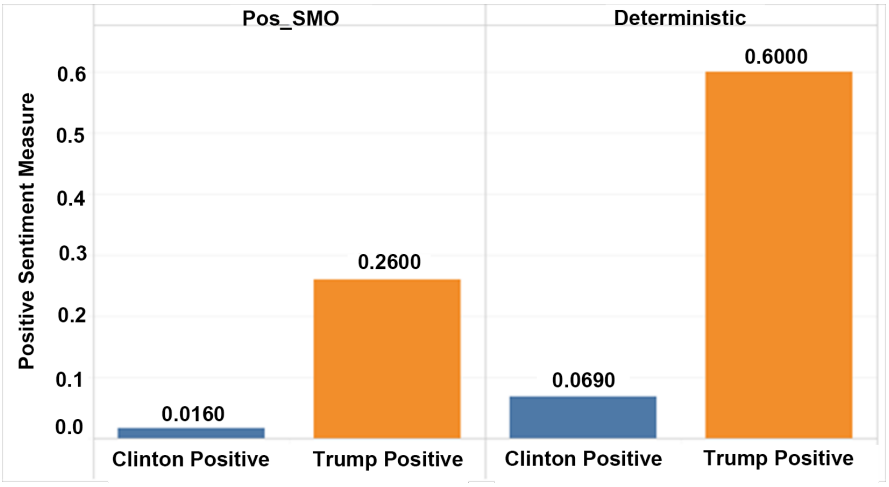
Our Deterministic Topic Model as discussed in 4.3 was applied to the November 2016 pre-election tweet streams. The positive polarity orientation for Donald Trump was increased to 0.60 while the positive polarity measure for Hillary Clinton was 0.069. From our results show in **Figure 3(b)** below, we witnessed the sharply increased positive sentiment orientation for candidate Donald Trump in the data streams during the pre-election November with candidate Donald Trump's volume of Trump-correlated topic tweets (53,009 tweets) compared to that for Hillary Clinton (8510 tweets) for Subjectivity Weighted *CMSE* shown in **Figure 3(a)**. Our system showed that Donald Trump as the definitive winner of the 2016 Presidential Election.

5.3. Cross Validation with Multinomial Naive Bayes Classifier for Deterministic and Probabilistic Models

Our cross validation was performed with the following experiment settings and an assumption that for a user chosen time period for a user given topic (event),



(a)



(b)

Figure 3. (a) Polarity comparison of two candidates: Clinton vs Trump with *CMSE* and subjectivity weighted *CMSE*; (b) Comparison of positive sentiment measure of two candidates with Pos_*SMO* and deterministic model.

data streams are collected from randomly chosen time frames, each in a 30 sec window, from the same social platform where the messages occur randomly for both candidates. To validate parallel stream data processing, we adopt the method of evaluation of big data stream classifier proposed in Bifet 2015 [7]. The standard K-fold cross-validation, which is used in other works with batch methods, treats each fold of the stream independently, and therefore may miss concept drift occurring in the data stream. To overcome these problems, we employed the strategy K-fold distributed cross-validation [7] to validate stream data. Assuming we have K different instances of the classifier, we want to evaluate running in parallel. The classifier does not need to be randomized. Each time a new example arrives, it is used in 10-fold distributed cross-validation: each example was used for testing in one classifier selected randomly, and used for training by all the others.

10 fold distributed cross validation were performed on our stream data

processing in each two different data splits. 60%: 40% training data: test data, and 90%: 10%. Average accuracy was taken for each split, for each deterministic and probabilistic model. Each cross validation was performed with classifier optimization parameters providing the model a variance of smoothing factors, features for term frequency and numeric values for min document frequency. Figure 4 illustrates the accuracies of deterministic and probabilistic models. 10 fold cross validation on 90%: 10% split with Deterministic model showed the highest accuracy with an average accuracy of 81% and the average accuracy of the Probabilistic model showed the almost comparable result with 80%. In comparison with the existing works, the overall average accuracy from the cross validation on each model shows 1% - 22% improvement from the previous work [6] [7] [8] [9] [22] [23] [24] [29] [30]. **Figure 4** below illustrates the cross validation results of the Deterministic and Probabilistic models.

6. Conclusions

The main contribution of this paper is the design and development of a real time big data stream analytic framework; providing a foundation for an infrastructure of real time sentiment analysis on big text streams. Our framework is proven to be an efficient, scalable tool to extract, score and analyze opinions on user generated text streams per user given topics in real time or near real time. The experiment results demonstrated the ability of our system architecture to accurately predict the outcome of the 2016 Presidential Race against candidates Hillary Clinton and Donald Trump.

The proposed fully analytic Deterministic and Probabilistic sentiment models coupled with the real-time streaming components were tested on the user tweet streams captured during pre-election month in October 2016 and the pre-election week of November 2016. The results proved that our system was

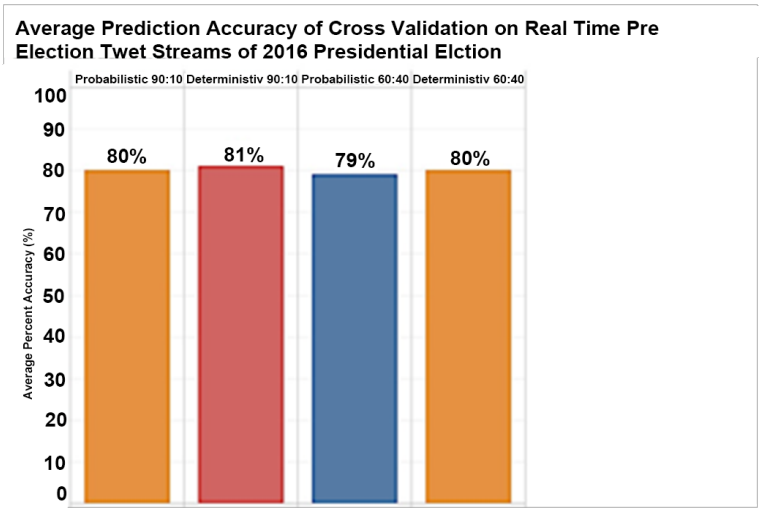


Figure 4. Average cross validation prediction accuracy on real time pre election tweet streams of 2016 presidential election for deterministic vs. probabilistic model.

able to predict correctly Donald Trump as the definitive winner of the 2016 Presidential election. The cross validation results showed that the Deterministic Topic Model in real time processing consistently improved the accuracy with average 81% and the Probabilistic Topic Model with average 80% compared to the accuracies of the previous works, ranging from 59% to 80%, in the literature [6] [7] [8] [9] [22] [23] [24] [29] [30] that lacked the complexity of sentiment analysis processing, either in batch or real time processing.

Finally, SMDSSAS performed efficient real-time data processing and sentiment analysis in terms of scalability. The system uses the continuous processing of a smaller window of data stream (e.g. consistent processing of a 30sec window of streaming data) in which machine learning analytics were performed on the context stream resulting in more accurate predictions with the ability of the system to continuously apply multi-layered fully analytic processes with complex sentiment models to a constant stream of data. The improved and stable model accuracies demonstrate that our proposed framework with the two sentiment models offers a scalable real-time sentiment analytic framework alternative for big data stream analysis over the traditional batch mode data analytic frameworks.

Acknowledgements

The research in this paper was partially supported by the Engineering College of CSU under the Graduate Research grant.

References

[1] Armbrust, M., Xin, R.S., Lian, C., Huai, Y., Liu, D., Bradley, J.K., Meng, X., Kaftan, T., Franklin, M.J., Ghodsi, A. and Zaharia, M. (2015) Spark SQL: Relational Data Processing in SPARK. *Proceedings of the ACM SIGMOD International Conference on Management of Data*, Melbourne, 31 May-4 June 2015, 1383-1394. https://doi.org/10.1145/2723372.2742797

[2] Sagiroglu, S. and Sinanc, D. (2013) Big Data: A Review. 2013 *International Conference on Collaboration Technologies and Systems* (*CTS*), San Diego, 20-24 May 2013, 42-47. https://doi.org/10.1109/CTS.2013.6567202

[3] Lars, E. (2015) What's the Best Way to Manage Big Data for Healthcare: Batch vs. Stream Processing? Evariant Inc., Farmington.

[4] Hu, M. and Liu, B. (2004) Mining and Summarizing Customer Reviews. *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Seattle, 22-25 August 2004, 168-177.

[5] Liu, B. (2010) Sentiment Analysis and Subjectivity. In: Indurkhya, N. and Damerauthe, F.J., Eds., *Handbook of Natural Language Processing*, 2nd Edition, Chapman and Hall/CRC, London, 1-38.

[6] Wang, H., Can, D., Kazemzadeh, A., Bar, F. and Narayanan, S. (2012) A System for Real-Time Twitter Sentiment Analysis of 2012 U.S. Presidential Election Cycle. *Proceedings of ACL* 2012 *System Demonstrations*, Jeju Island, 10 July 2012, 115-120.

[7] Bifet, A., Maniu, S., Qian, J., Tian, G., He, C. and Fan, W. (2015) StreamDM: Advanced Data Mining in Spark Streaming. *IEEE International Conference on Data*

Mining Workshop (*ICDMW*), Atlantic City, 14-17 November 2015, 1608-1611.

[8] Kulkarni, S., Bhagat, N., Fu, M., Kedigehalli, V., Kellogg, C., Mittal, S., Patel, J.M., Ramasamy, K. and Taneja, S. (2015) Twitter Heron: Stream Processing at Scale. *Proceedings of the* 2015 *ACM SIGMOD International Conference on Management of Data*, Melbourne, 31 May-4 June 2015, 239-250. https://doi.org/10.1145/2723372.2742788

[9] Nair, L.R. and Shetty, S.D. (2015) Streaming Twitter Data Analysis Using Spark For Effective Job Search. *Journal of Theoretical and Applied Information Technology*, **80**, 349-353.

[10] Pang, B. and Lee, L. (2004) A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts. *Proceedings of the* 42*nd Annual Meeting on Association for Computational Linguistics*, Barcelona, 21-26 July 2004, 271-278. https://doi.org/10.3115/1218955.1218990

[11] Harris, Z. (1954) Distributional Structure Word. *WORD*, **10**, 146-162. https://www.tandfonline.com/doi/abs/10.1080/00437956.1954.11659520

[12] Blei, D., Ng, A. and Jordan, N. (2003) Latent Dirichlet Allocation. *Journal of Machine Learning Research*, **3**, 993-1022.

[13] Zhai, C. and Lafferty, J. (2004) A Study of Smoothing Methods for Language Models Applied to Information Retrieval. *ACM Transactions on Information Systems*, **22**, 179-214.

[14] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. and Dean, J. (2013) Distributed Representations of Words and Phrases and Their Compositionality. *Proceedings of the* 26*th International Conference on Neural Information Processing Systems*, Lake Tahoe, 5-10 December 2013, 3111-3119

[15] Mikolov, T., Chen, K., Corrado, G. and Dean, J. (2013) Efficient Estimation of Word Representations in Vector Space. *Proceedings of Workshop at ICLR*, Scottsdale, 2-4 May 2013, 1-11.

[16] Dai, A., Olah, C. and Le, Q. (2015) Document Embedding with Paragraph Vectors. arXiv:1507.07998.

[17] Le, Q. and Mikolov, T. (2014) Distributed Representations of Sentences and Documents. *Proceedings of the* 31*st International Conference on Machine Learning* (*ICML-14*), Beijing, 21-26 June 2014, II1188-II1196.

[18] Firth, J.R. (1930) A Synopsis of Linguistic Theory 1930-1955. In: Firth, J.R., Ed., *Studies in Linguistic Analysis*, Longmans, London, 168-205.

[19] Tang, J., Qu, M. and Mei, Q.Z. (2015) PTE: Predictive Text Embedding through Large-Scale Heterogeneous Text Networks. arXiv:1508.00200.

[20] Bo, P. and Lee, L. (2008) Opinion Mining and Sentiment Analysis. In: de Rijke, M., *et al.*, Eds., *Foundations and Trends® in Information Retrieval*, James Finlay Limited, Ithaca, 1-135. https://doi.org/10.1561/1500000011

[21] Maite, T., Brooke, J., Tofiloski, M., Voll, K. and Stede, M. (2011) Lexicon-Based Methods for Sentiment Analysis. *Computational Linguistics*, **37**, 267-307. https://doi.org/10.1162/COLI_a_00049

[22] O'Connor, B., Balasubramanyan, R., Routledge, B. and Smith, N. (2010) From Tweets to Polls: Linking Text Sentiment to Public Opinion Time Series. *Proceedings of the International AAAI Conference on Weblogs and Social Media* (*ICWSM* 2010), Washington DC, 23-26 May 2010, 122-129.

[23] Cheng, K.M.O. and Lau, R. (2015) Big Data Stream Analytics for Near Real-Time Sentiment Analysis. *Journal of Computer and Communications*, **3**, 189-195. https://doi.org/10.4236/jcc.2015.35024

[24] Cheng, K.M.O. and Lau, R. (2016) Parallel Sentiment Analysis with Storm. *Transactions on Computer Science and Engineering*, 1-6.

[25] Thusoo, A., Sarma, J.S., Jain, N., Shao, Z., Chakka, P., Zhang, N., Antony, S., Liu, H. and Murthy, R. (2010) Hive—A Petabyte Scale Data Warehouse Using Hadoop. *Proceedings of the International Conference on Data Engineering*, Long Beach, 1-6 March 2010, 996-1005.

[26] Manning, C., Surdeanu, A., Bauer, J., Finkel, J., Bethard, S. and McClosky, D. (2014) The Stanford CoreNLP Natural Language Processing Toolkit. *Proceedings of* 52*nd Annual Meeting of the Association for Computational Linguistics*: *System Demonstrations*, Baltimore, 23-24 June 2014, 55-60. https://doi.org/10.3115/v1/P14-5010

[27] Finkel, J., Grenager, T. and Manning, C. (2005) Incorporating Non-Local Information into Information Extraction Systems by Gibbs Sampling. *Proceedings of the* 43*rd Annual Meeting of the Association for Computational Linguistics* (*ACL* 2005), Ann Arbor, 25-30 June 2005, 363-370.

[28] Wilson, T., Wiebe, J. and Hoffman, P. (2005) Recognizing Contextual Polarity in Phrase-Level Sentiment Analysis. *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, Vancouver, 6-8 October 2005, 347-354. https://doi.org/10.3115/1220575.1220619

[29] Wang, S., Zhiyuan, C. and Liu, B. (2016) Mining Aspect-Specific Opinion Using a Holistic Lifelong Topic Model. *Proceedings of the* 25*th International Conference on World Wide Web*, Montréal, 11-15 April 2016, 167-176. https://doi.org/10.1145/2872427.2883086

[30] Yi, J., Nasukawa, T., Bunescu, R. and Niblack, W. (2003) Sentiment Analyzer: Extracting Sentiments about a Given Topic Using Natural Language Processing Techniques. *Proceedings of IEEE International Conference on Data Mining* (*ICDM*), Melbourne, 22-22 November 2003, 1-8. https://doi.org/10.1109/ICDM.2003.1250949

[31] Tilve, A. and Jain, S. (2017) A Survey on Machine Learning Techniques for Text Classification. *International Journal of Engineering Sciences and Research*, **6**, 513-520.

[32] Rennie, J., Shih, L., Teevan, J. and Karger, D. (2003) Tackling the Poor Assumptions of Naive Bayes Text Classifiers. *Proceedings of the Twentieth International Conference on International Conference on Machine Learning*, Washington DC, 21-24 August 2003, 616-623.