8-2016

Feature Extraction to Improve Nowcasting Using Social Media Event Detection on Cloud Computing and Sentiment Analysis

David L. Kimmey

Indiana University - Purdue University Fort Wayne

Follow this and additional works at: http://opus.ipfw.edu/masters_theses

Part of the Other Computer Sciences Commons, and the Theory and Algorithms Commons

Recommended Citation


This Master’s Research is brought to you for free and open access by the Graduate Student Research at Opus: Research & Creativity at IPFW. It has been accepted for inclusion in Masters’ Theses by an authorized administrator of Opus: Research & Creativity at IPFW. For more information, please contact admin@lib.ipfw.edu.
This is to certify that the thesis/dissertation prepared

By  David L. Kimmey

Entitled
FEATURE EXTRACTION TO IMPROVE NOWCASTING USING SOCIAL MEDIA EVENT DETECTION ON CLOUD COMPUTING AND SENTIMENT ANALYSIS

For the degree of  Master of Science

Is approved by the final examining committee:

Jin Soung Yoo  Chair

Approved by Major Professor(s):  Jin Soung Yoo

Approved by:  Jin Soung Yoo  7/5/2016

Head of the Departmental Graduate Program  Date
FEATURE EXTRACTION TO IMPROVE NOWCASTING USING SOCIAL MEDIA
EVENT DETECTION ON CLOUD COMPUTING AND SENTIMENT ANALYSIS

A Thesis
Submitted to the Faculty
of
Purdue University
by
David L. Kimmey

In Partial Fulfillment of the
Requirements for the Degree
of
Master of Science

August 2016
Purdue University
Fort Wayne, Indiana
This thesis is dedicated to my fiancée Sonia McDaniel.

Thank you for your inspiration, support, patience, and love.
ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to all those who made this thesis possible. A special thanks to Dr. Jin Soung Yoo, my committee chair and advisor. Thank you for allowing me to assist in your research and publications. It was an honor working with you on the Air Force Summer Faculty Fellowship Program, and working as your Graduate Research and Teaching Assistant. I would like to thank IPFW for awarding me the Graduate Research Assistantship position. I would also like to thank my committee members for their expertise, feedback, and support: Dr. David Liu, Dr. Zesheng Chen, and Dr. Anyi Liu. I wish to thank Barbara Lloyd for her formatting expertise, and Shannon Johnson for her bibliography help. I would also like to extend a special thanks to Dr. Peter Ng—thank you for years of advice, support, and encouragement.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>LIST OF TABLES</td>
<td>vii</td>
</tr>
<tr>
<td>LIST OF FIGURES</td>
<td>viii</td>
</tr>
<tr>
<td>ABSTRACT</td>
<td>ix</td>
</tr>
<tr>
<td>CHAPTER 1. INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>CHAPTER 2. BACKGROUND AND PROBLEM STATEMENTS</td>
<td>9</td>
</tr>
<tr>
<td>2.1 Social Media Event Detection</td>
<td>9</td>
</tr>
<tr>
<td>2.1.1 Background Concept</td>
<td>9</td>
</tr>
<tr>
<td>2.1.2 Problem Statement</td>
<td>11</td>
</tr>
<tr>
<td>2.2 Nowcasting</td>
<td>13</td>
</tr>
<tr>
<td>2.2.1 Background Concept</td>
<td>13</td>
</tr>
<tr>
<td>2.2.2 Problem Statement</td>
<td>14</td>
</tr>
<tr>
<td>CHAPTER 3. RELATED WORKS</td>
<td>15</td>
</tr>
<tr>
<td>3.1 Social Media Event Detection</td>
<td>15</td>
</tr>
<tr>
<td>3.1.1 Burst Method</td>
<td>15</td>
</tr>
<tr>
<td>3.1.2 Anomaly Detection Method</td>
<td>16</td>
</tr>
<tr>
<td>3.1.3 Probabilistic Soft Logic Method</td>
<td>17</td>
</tr>
<tr>
<td>3.2 Sentiment Analysis</td>
<td>17</td>
</tr>
<tr>
<td>3.2.1 Natural Language Processing</td>
<td>17</td>
</tr>
<tr>
<td>3.3 Nowcasting</td>
<td>18</td>
</tr>
<tr>
<td>3.3.1 Time Series with Google Trends</td>
<td>18</td>
</tr>
<tr>
<td>3.3.2 Algorithmic Models</td>
<td>19</td>
</tr>
<tr>
<td>3.3.3 Early Detection of Influenza</td>
<td>20</td>
</tr>
</tbody>
</table>
3.3.4 Lightweight Methods to Estimate Influenza Rates and Alcohol Sales Volume from Twitter Messages ................................................................. 20

CHAPTER 4. SOCIAL MEDIA EVENT DETECTION ........................................... 22
4.1 Social Media Event Detection ................................................................................ 22
4.1.1 K-SPRE ....................................................................................................... 22
4.1.2 Social Media Event Detection (SMED) Algorithm ....................................... 26
4.2 Social Media Event Detection Method on a Cloud Computing Environment ...... 27
4.2.1 MapReduce and SpatialHadoop .................................................................. 27
4.2.2 Parallel Social Media Event Detection (PSMED) Algorithm ....................... 29

CHAPTER 5. SENTIMENT ANALYSIS ............................................................ 33
5.1 Sentiment Analysis Method ................................................................................. 33
5.2 Sentiment Analysis Algorithm ............................................................................ 33

CHAPTER 6. NOWCASTING MODEL .............................................................. 35
6.1 Nowcasting Model ............................................................................................. 35
6.1.1 Artificial Neural Network (ANN) ............................................................... 35
6.2 Evaluation Methods ........................................................................................... 37
6.2.1 Markov First-Order Autoregressive Scheme Methodology .......................... 37
6.2.2 Evaluation Methodologies ........................................................................... 37

CHAPTER 7. CASE STUDY AND FRAMEWORK ............................................ 39
7.1 Case Study Description ....................................................................................... 39
7.2 General KDD Framework ................................................................................. 39
7.3 Framework for Nowcasting with Case Study .................................................... 40
7.3.1 Data Selection Phase 1 ................................................................................ 42
7.3.2 Cleaning & Integration Phase 2 .................................................................... 44
7.3.3 Reduction Phase 3 ....................................................................................... 46
7.3.4 Transformation & Discretization Phase 4 .................................................... 47
7.3.5 Regression Phase 5 ..................................................................................... 48
7.3.6 Evaluation Phase 6 ...................................................................................... 50

CHAPTER 8. EXPERIMENTAL EVALUATION ............................................... 52
8.1 Experimental Setup ........................................................................................... 52
# LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1 Sample of social media logs</td>
<td>5</td>
</tr>
<tr>
<td>1.2 Aggregated social media logs with the CDC's weekly ILI report value (Y)</td>
<td>5</td>
</tr>
<tr>
<td>4.1 Social media instances</td>
<td>25</td>
</tr>
<tr>
<td>7.1 Sample (a) and (b) of the CDC's data set attributes for influenza like illness</td>
<td>44</td>
</tr>
<tr>
<td>7.2 Sample of queried and cleaned Twitter data</td>
<td>45</td>
</tr>
<tr>
<td>7.3 Sample of cleaned Twitter data, keyword frequency</td>
<td>45</td>
</tr>
<tr>
<td>7.4 Sample of cleaned CDC data</td>
<td>45</td>
</tr>
<tr>
<td>7.5 Sample (a) and (b) output attributes from Social Media Event Detection (SMED)</td>
<td>45</td>
</tr>
<tr>
<td>7.6 Sample output of sentiment analysis</td>
<td>45</td>
</tr>
<tr>
<td>7.7 Sample of the nowcast model's training data set for week 8</td>
<td>45</td>
</tr>
<tr>
<td>7.8 Sample of the nowcast model's test data set (i.e., prediction) for week 8</td>
<td>45</td>
</tr>
<tr>
<td>7.9 Sample of the nowcast model's training data set for week 9</td>
<td>50</td>
</tr>
<tr>
<td>7.10 Sample of the nowcast model's test data set (i.e., prediction) for week 9</td>
<td>50</td>
</tr>
<tr>
<td>8.1 AR(1) and nowcast model statistical error measurements</td>
<td>56</td>
</tr>
</tbody>
</table>
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1 Correlation model between social media weekly frequencies and the CDC's ILI weekly report value (Y)</td>
<td>5</td>
</tr>
<tr>
<td>1.2 Forecast of the CDC's week 10 ILI report value (Y)</td>
<td>6</td>
</tr>
<tr>
<td>2.1 Example outputs from 3 different methods used to preprocess social media for an influenza event</td>
<td>10</td>
</tr>
<tr>
<td>4.1 Plot of K-SPRE, where k = 2</td>
<td>23</td>
</tr>
<tr>
<td>4.2 Plot of social media instances</td>
<td>25</td>
</tr>
<tr>
<td>6.1 Artificial neural network with back propagation</td>
<td>36</td>
</tr>
<tr>
<td>7.1 Activities diagram of Knowledge Discovery from Data (KDD) [43]</td>
<td>40</td>
</tr>
<tr>
<td>7.2 Nowcasting with social media—first four framework phases</td>
<td>40</td>
</tr>
<tr>
<td>7.3 Nowcasting with social media—last two framework phases</td>
<td>40</td>
</tr>
<tr>
<td>7.4 Single raw Twitter instance (i.e., Tweet)</td>
<td>40</td>
</tr>
<tr>
<td>8.1 AR(1) model forecasts and CDC's % weighted influenza like illness values</td>
<td>56</td>
</tr>
<tr>
<td>8.2 Nowcast model forecasts and CDC's % weighted influenza like illness values</td>
<td>56</td>
</tr>
<tr>
<td>8.3 Normalized time improvement of parallel social media event detection (PSMED) algorithm distributed over an AWS cluster</td>
<td>57</td>
</tr>
<tr>
<td>8.4 Time improvement of parallel social media event detection (PSMED) algorithm distributed over an AWS cluster</td>
<td>58</td>
</tr>
</tbody>
</table>
ABSTRACT

Kimmey, David L. M.S., Purdue University, August 2016. Feature Extraction to Improve Nowcasting Using Social Media Event Detection on Cloud Computing and Sentiment Analysis. Major Professor: Jin Soung Yoo.

Nowcasting is defined as the prediction of the present, the very near future, and the very recent past using real-time data. Nowcasting with social media creates challenges because of the HACE characteristics of big data (i.e., heterogeneous, autonomous, complex, and evolving associations). Thus, this thesis proposes a feature extraction method to improve nowcasting with social media. The proposed social media event detection algorithm utilizes K-SPRE methodology and the results are processed with sentiment analysis. In addition, we develop a parallel algorithm of the methodology on a cloud environment, and we adapt an artificial neural network to build a predictive model for nowcasting. Furthermore, we complete a case study with real data: Twitter and the Center for Disease Control (CDC) influenza like illness (ILI) reports. Experiments with predicting the CDC’s ILI report shows nowcasting with social media outperforms the traditional time series AR(1) model by as much as 16% to 20%, in terms of statistical error. In addition, implementation of the social media event detection algorithm with cloud computing improved the algorithm’s running time by 65%.
CHAPTER 1. INTRODUCTION

Nowcasting with social media is a hot topic of interest to businesses, central banks, and government agencies. Nowcasting with social media assists these entities and agencies in understanding public opinion and trends; in creating timely forecasts of economic indicators; and in forecasting early detection of disease activity—thereby allowing rapid disease response, which reduces the public impact of disease [1, 2]. Nowcasting uses real-time data and is defined as the prediction of the present, the very near future and the very recent past; furthermore, the term is a contraction of now and forecasting, and has been used for a long time in meteorology and recently also in economics [3]. Moreover, nowcasting is seen as a form of contemporaneous forecasting or predicting the present [1].

Social media is a source of real-time data that individuals create and voluntarily share on major social media generators such as Facebook, Twitter, Google, Yahoo, and Instagram. Nowcasting events and social media analysis are growing areas of research that have advanced significantly as social media is becoming more popular [4]. The following are examples of how nowcasting and social media are being used together:

- Geo-economic events: how the public’s sentiment affects the stock market [1, 5].
• Discovery of unusual social events: the discovery of demonstrations, spontaneous festivals, and natural disasters—earthquakes and storms [6-10].

• Geographic disease and influenza trends: the tracking and monitoring of disease and influenza [2, 11, 12].

• Social questions: are popular events associated with increased public sentiment [13]?

• Predicting political alignment: management of political strategy [14].

To make timely decisions, governments and businesses need to forecast, in real time, trends and events which may affect their operations. The question is: how does the business or government agency obtain clean, relevant, and timely data to complete the nowcasting process for their specific reporting needs?

One possible solution is to query real-time data from social media. This solution would allow them to nowcast a trend or event; however, the defining and detection of events (i.e. feature types) has long been a research topic and is a non-trivial task. In addition, social media has heterogeneous, autonomous, complex, and evolving associations—the HACE characteristics of Big Data [15]. Thus, this solution must overcome obstacles such as managing the HACE characteristics of social media.

This work developed a feature extraction method to improve nowcasting with social media: K-SPRE (k-spree). K-SPRE is the combination of four methodologies: (1) k-nearest-neighbor, (2) social media spatial relations, (3) probabilistic soft logic, and (4) referenced events, which is used in both the SMED and PSMED algorithms.
The proposed social media event detection (SMED) algorithm utilizes K-SPRE methodology and the results are processed with sentiment analysis. In short, the SMED algorithm finds user specified events within the social media by identifying the relationships between geospatial and causal model variables within the data set. Furthermore, the SMED algorithm returns the associated social media content, geospatial locations, and time aspects from the discovered specified event. Thus, the SMED algorithm provides three advantages over the query approach: (1) the user specified event’s corpus has less noise; (2) clearer recognition of the user specified event’s approximate current location, and (3) identification of popular keyword themes from a user’s specified event within social media. Moreover, this work uses sentiment analysis to index the corpus of each social media instance within the user specified event. This derived feature is a measurement of the public’s sentiment (i.e., strength or weakness) about the specified event. In addition, we develop a parallel version of the SMED algorithm, named PSMED, on a cloud environment and adapt an artificial neural network to build a predictive model for nowcasting using the extracted features.

This work uses a social media data source (e.g., Twitter data) to nowcast a corporate or government agency report (e.g., the CDC’s influenza like illness report), and to compare and contrast the nowcast results with a traditional time series model. There are three advantages to using nowcasting to predict report values. First, research has shown high correlations between social media data and government reports of 76.7% to 90% [4]. Second, it’s not trivial to forecast turning points in time series data, however as seen in Figure 1.2, a forecast using nowcasting has turning points with higher correlation to the actual data than a traditional time series model [1]. And third, many government
and business report values do not use real-time data values (i.e., the report’s value (Y) describes past results from the previous week, month, or quarter), thus, nowcasting provides a real-time data prediction solution for real-time decisions.

Table 1.1 and 1.2, and Figure 1.1 and 1.2 illustrate a mock example of nowcasting with social media; specifically the prediction of week 10 of the CDC’s weekly influenza like illness (ILI) report value (Y), as shown in Figure 1.2. This example is only intended as a simple example to help readers build intuition about the nowcast process, and does not meet statistical requirements. Table 1.1 shows a collection of social media logs queried from the keyword flu with two attributes: media, and week. Furthermore, the media attribute is aggregated each week, counted, labeled as flu instances, and is combined with the existing CDC ILI reported values (Y), as shown in Table 1.2. Next, Table 1.2 is used to create the correlation model between the frequency of weekly flu instances and the CDC’s ILI weekly reported values (Y), as shown in Figure 1.1. And finally, the correlation model is used to nowcast week 10 of the CDC’s ILI report value (Y) in Figure 1.2. As illustrated, the time series model incorrectly forecasted a continued uptrend in the ILI report value, whereas, the nowcast model correctly forecasted a downturn, and was validated by the CDC’s actual report value (Y), as shown in Figure 1.2.
Table 1.1 Sample of social media logs

<table>
<thead>
<tr>
<th>Social Media Logs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Media</td>
</tr>
<tr>
<td>&quot;I am sick&quot;</td>
</tr>
<tr>
<td>&quot;I feel ill&quot;</td>
</tr>
<tr>
<td>&quot;I need flu medicine&quot;</td>
</tr>
<tr>
<td>&quot;I need a doctor&quot;</td>
</tr>
<tr>
<td>&quot;she has the flu&quot;</td>
</tr>
<tr>
<td>&quot;we have the flu&quot;</td>
</tr>
<tr>
<td>Week 7</td>
</tr>
<tr>
<td>Week 8</td>
</tr>
<tr>
<td>Week 8</td>
</tr>
<tr>
<td>Week 8</td>
</tr>
<tr>
<td>Week 9</td>
</tr>
<tr>
<td>Week 9</td>
</tr>
</tbody>
</table>

Table 1.2 Aggregated social media logs with the CDC's weekly ILI report value (Y)

<table>
<thead>
<tr>
<th>Aggregated Social Media Logs with CDC ILI Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Week</td>
</tr>
<tr>
<td>Number of Weekly Flu Instances</td>
</tr>
<tr>
<td>----------------------------------</td>
</tr>
<tr>
<td>Week 7</td>
</tr>
<tr>
<td>Week 8</td>
</tr>
<tr>
<td>Week 9</td>
</tr>
</tbody>
</table>

Figure 1.1 Correlation model between social media weekly frequencies and the CDC's ILI weekly report value (Y)
Figure 1.2 Forecast of the CDC's week 10 ILI report value (Y)

There are several sources of social media, and the data from all of these sources have challenges due to the HACE characteristics of big data (i.e., heterogeneous, autonomous, complex, and evolving associations) [15]. For example, although the defining and detection of events for data mining topics (i.e. feature types) has long been a research topic, the HACE characteristics of Twitter make defining and detection of Twitter events a non-trivial task. In short, there are five main challenges: (1) Twitter’s tweet content is complex, heterogeneous, and is usually overwhelmed with “babble” (i.e., about 40% of tweets queried or data mined will not include the queried keyword); (2) Twitter event detection algorithms need to be scalable given the sheer amount of tweets; (3) Twitter’s broadcasted tweet locations are autonomous and have evolving association that are dynamically changing and increasing in a real time nature; (4) because Twitter’s API only grants access to a 1% sample of the Twitter data, tweets queried or data mined by keywords can result in sparse geospatial data sets; and (5) concerns about Twitter’s sampling strategy and the quality of the data has been raised [9, 10, 16, 17].
Furthermore, nowcasting adds an additional layer of complexity and challenge. For example, the use of a correlation model to nowcast a report value (Y) deals with the relationship between dependent and independent variables (e.g., between social media and the CDC’s weekly ILI report value (Y)), and the correlation model does not necessarily imply causation [18]. Thus, causality must be justified, or inferred, from a theory that underlies the phenomenon that is being empirically tested [18]. If the original data cannot be mined from the social media, then the parameters of an underlying theory that support the correlated relationship should be data mined as a proxy for the original data. This will produce a robust, and more accurate, correlation model.

The following example illustrates how a government report ‘Initial Claims’, which is not a word commonly found in social media, can still be nowcasted by data mining for substitute words that work as a proxy. ‘Initial Claims’ have a record of being a good leading indicator for the U.S. economy [1]. However, mining social media for ‘Initial Claims’ would result in a very sparse data set (i.e., few individuals would use the words ‘Initial Claims’ in their social media content). Thus, using the theory that it is natural to expect an unemployed person to look for work, one might mine social media using different keywords as a proxy for ‘Initial Claims’, such as: ‘file for unemployment’, ‘unemployment office’, ‘unemployment benefits’, ‘unemployment claims’, ‘jobs’, and ‘resume’ [1].
Considering the previous challenges, the thesis contributions are as follows:

- We propose a social media event detection (SMED) algorithm, which utilizes K-SPRE methodology to find user specified events within social media.
- We develop a parallel and distributed version of the SMED algorithm on MapReduce resulting in the PSMED algorithm.
- We present a framework for nowcasting (i.e., predicting the present) with social media, and we incorporate causality by employing user defined causal model variables.
- We conduct experiments and validate the nowcasting framework on real data using the popular social media data source—Twitter.

The remainder of this thesis is organized as follows. Chapter 2 describes the background and problem statements for social media event detection and nowcasting. Chapter 3 describes the related works for social media event detection, sentiment analysis, and nowcasting. Chapter 4 presents K-SPRE, the social media event detection (SMED) algorithm, and the parallel social media event detection (PSMED) algorithm. Chapter 5 describes the sentiment analysis method and algorithm used in this work. Chapter 6 describes the methodologies used for nowcasting. Chapter 7 presents a case study and framework using Twitter and CDC data. Chapter 8 presents the experimental setup and evaluation, and in chapter 9 we conclude this thesis.
CHAPTER 2. BACKGROUND AND PROBLEM STATEMENTS

Chapter 2 describes the background and problem statements for detecting events in social media, sentiment analysis, and nowcasting.

2.1 Social Media Event Detection

2.1.1 Background Concept

Social media contains several events (i.e., a thing(s) that happen, especially one of importance) not bound by a known geography. Nonetheless, if we know an event’s current location, the event can be mined using social media from the specific location. However, in today’s dynamic world, we do not always know where an event will occur (e.g., influenza outbreaks, weather events—tornados and hurricanes, and economic shocks).

Hence, to identify events in social media without a geographic location can be a non-trivial task. For instance, a naïve query of social media for an event with one keyword (e.g., flu) will return data randomly sampled from many different locations. As a result, the majority of social media returned is not from the event’s unrevealed location as in Figure 2.1 (a). Therefore, the flu event’s approximate current geographic region is not easily identified. In addition, if the social media is queried with multiple keywords
(e.g., flu and sick and headache and chills and fever) the returned data will be sparse and randomly distributed as in Figure 2.1 (b). Ultimately, this is because few individuals will include all the keywords in a single piece of social media content.

Consequently, this work developed an event detection algorithm—Social Media Event Detection (SMED), which finds a user specified event within the social media’s content and queried region. SMED identifies the relationships between the geospatial and causal model variables within the data set. The subsequent data contains a higher density of social media from the specified event’s unrevealed location. Furthermore, post processing of the SMED data identifies an approximate current location of the user specified event, Figure 2.1 (c).

SMED provides three advantages over the naïve query approach: (1) the user specified event’s corpus has less noise; (2) clearer recognition of the user specified event’s approximate current location, as shown in Figure 2.1 (c); and (3) identification of popular keyword themes from a user’s specified event within social media content.

Figure 2.1 Example outputs from 3 different methods used to preprocess social media for an influenza event.
Notice: (c) SMED locates more points within the flu event’s unrevealed location (providing an approximate current location).
2.1.2 Problem Statement

A social media event is an event that is shared through social media, and thus, has a location, time sensitivity aspects, and shared media content.

Definition 2.1.2.1 Event: Given a set of geospatial objects with social media content $g_i \in G$ found between time $t_1$ to time $t_2$, and a set of causal variables $\{x_1, x_2, ..., x_n\} \in C$, then an event $E$ is defined as the set of relationships $R$ from time $t_1$ to time $t_2$ (i.e., frequency of relationships). The individual relationship $r_i \in R$ is a probabilistic soft logic (PSL) rule, where $E = \{g_i, x_j \in G \cap C \mid R(g_i, x_j) = TRUE\}$, where the individual relationship $r_i \in R$ is a PSL rule of the form

$$RW: trigger\_event_1(w_1) \land \ldots \land trigger\_event_n(w_n) \Rightarrow final\_event_{n+1}(w_{n+1}),$$

where the rule weight $RW$ is the average distance between the rule’s trigger-event causal variables and final-event causal variable, normalized in the range from 0 to 1, and has a given threshold (e.g., $RW \geq 0.70$); and $w$ is the probability of the individual trigger-event causal variable or the individual final-event causal variable.

Consider the following mock example with three rules:

- 0.97: nausea(0.33) $\land$ fever(0.56) $\Rightarrow$ flu (0.11),
- 0.80: nausea(0.23) $\land$ sick(0.46) $\Rightarrow$ flu (0.31),
- 0.68: headache(0.43) $\land$ fever(0.47) $\Rightarrow$ flu (0.10)

then the flu event would be defined by the first two PSL rules with $RW \geq 0.70$. 
The problem statement of detecting events in social media can be described as follows.

Given:

(1) A final-event causality variable $x_1$ (i.e., a user specific event type)

(2) A set of trigger-event causality variables $\{x_2, \ldots, x_n\}$

(3) A data set of social media objects $S$, where each object $s_i \in S$ has a vector of information $\{social\ media\ key\ word, location(latitude, longitude), date\ created, and\ social\ media\ content\}$

Build:

A parallel social media event detection (PSMED) algorithm to find a user specified event within social media.

Objective:

To find and locate a user specified event within social media for use as attributes in a nowcast model.

Constraints:

The availability of probabilistic soft logic (PSL) rules with a given threshold (e.g., RW $\geq 0.70$).
2.2 Nowcasting

2.2.1 Background Concept

The first step in nowcasting with social media is to define the causal model (i.e., define the relationship underlying the data) being used as theoretical support for the nowcast model [18].

A causal model is a mathematical object that assigns truth values to sentences involving causal and counterfactual relationships [19].

For example, the causal model is a triple \( M = < T, V, F > \) where: (i) \( U \) is a set of variables, called exogenous (i.e., variables that a model takes as given), that are determined by factors outside the model; (ii) \( V \) is a set \( \{ V_1, V_2, \ldots, V_n \} \) of variables, called endogenous (i.e., variables that a model tries to explain, traditionally \( Y \)), that are determined by variables in the model, namely, variables in \( U \cup V \); (iii) \( F \) is a set of functions \( \{ f_1, f_2, \ldots, f_n \} \) where each \( f_i \) is a mapping from \( U \times (V \setminus V_i) \) to \( V_i \). In other words, each \( f_i \) tells us the value of \( V_i \) given the values of all other variables in \( U \cup V \) [19].

**Definition 2.2.1.1 Nowcast Model**: Given causal model variables \( \{ x_1, x_2, \ldots, x_{p-1} \} \in X \subseteq (U \cup (V \setminus V_i)) \), a report value \( (Y) \), and a query function for social media data \( Q( ) \), then the report value \( (Y) \) is modeled by the regression form \( Y = B_0 + B_1 Q(x_1) + B_2 Q(x_2) + \ldots + B_{p-1} Q(x_{p-1}) + \epsilon \) using causal model variables, and is used to predict government or business report values [1-3, 20].
2.2.2 Problem Statement

The problem statement for nowcasting can be described as follows.

Given:
(1) A social media data set
(2) A set of causal variables \( \{x_1, x_2, ..., x_n\} \in X \subseteq (U \cup (V \backslash V_i)) \)
(3) A report value \( Y \)
(4) A social media query function \( Q( ) \)

Model:

The report value \( (Y) \) by the regression form

\[
Y = B_0 + B_1Q(x_1) + B_2Q(x_2) + ... + B_{p-1}Q(x_{p-1}) + \epsilon
\]

Objective:

Compare nowcast model predictions to time series model predictions using government or business report values for \( (Y) \).

Constraints:

The availability to query or mine causal model variables within the social media content.
CHAPTER 3. RELATED WORKS

3.1 Social Media Event Detection

The related works include the following three event detection categories: burst method, anomaly detection method, and probabilistic soft logic method.

3.1.1 Burst Method

The burst method detects an event based on the number of keywords showing an increase in count (i.e., burst). Yang, et al. [21] investigates text retrieval and clustering techniques based on the burst method. The paper admits a content-focused query works well; however, it does not work well for generic queries such as ‘What happened?’ or ‘What’s new?’ Furthermore, Yang, et al. considers event detection as a discovery problem for which data mining new patterns in document content can be utilized. In order not to confuse event detection with topic detection, Yang, et al. defines an event as something (non-trivial) happening in a certain place at a certain time. In addition, Yang, et al. applies hierarchical and incremental non-hierarchical clustering algorithms with a focus on combining context information and temporal patterns for event distribution. Yang, et al. research interests include: semantic and temporal properties of events; document clustering based on content and temporal adjacency (rather than just content);
event detection based on similarity versus novelty, and evaluation methods for retrospective and on-line detection. The paper’s results show, if an event is well defined and content information is properly used with temporal information for retrospective and online event detection, then basic techniques such as document clustering can be very effective. However, online event detection is more difficult than retrospective event detection [21].

3.1.2 Anomaly Detection Method

The anomaly detection method detects an event based on statistical outliers. Specifically, Lee, et al. [8] uses a boxplot for its simplicity and visualization of the data distribution (i.e., the minimum, the lower quartile, the median, the upper quartile, and the maximum sample statistics). In short, Lee, et al. detects real-world events by finding temporal and geographic irregularities in the writing and movements of tweets in a specific bounded range. The paper’s process follows three steps: (1) collecting geotagged tweets by a unique Twitter monitoring system, (2) identifying socio-graphic boundaries of Twitter users and measuring geographical regularities of crowd behavior, and (3) detecting geo-social events through a comparison to the regularities. Lee, et al. reported satisfactory performance even with a small number of test events. The process found 32/50 (62%) of expected events. In addition, unexpected social and natural events were found (e.g., a stadium baseball game and a sudden thunderstorm were found) [8].
3.1.3 Probabilistic Soft Logic Method

The probabilistic soft logic (PSL) method detects an event based on spatio-temporal and probabilistic relationships. Santos, et al. [22] developed this approach by using storytelling with spatio-logical inference. In short, the paper’s approach takes the probability of prior occurrences of trigger events, along with their spatial distances, as inputs and calculates their soft truths to forecast a final event. Santos, et al. uses a key idea that social events tend to be associated to other spatially and temporally-related nearby activates that can be used to uncover a final event. For example, observe(police, protesters) ∧ push(protesters, crowd) ⇒ cause(crowd, riot). Santos, et al. concluded their proposed approach provided significantly higher precision and higher recall scores than traditional probabilistic methods. Furthermore, the PSL rules with the lowest distance had the best forecasts [22].

3.2 Sentiment Analysis

3.2.1 Natural Language Processing

Sentiment analysis, or opinion mining, is the computational study of opinions, sentiments, and emotions expressed in text [23]. A paper by Blair-Goldensohn, et al. [24] presents a system that summarizes the sentiment of online reviews. The paper focuses on aspect-based summarization (ABS) models. The advantage of ABS models is the use of user provided labels and domain knowledge to increase the quality of the sentiment classification. Blair-Goldensohn, et al. suggests three main steps in their system: (1)
identify all sentiment laden text fragments in the online reviews; (2) identify relevant aspects of the reviews that are mentioned in the fragments; (3) aggregate sentiment over each aspect based on sentiment of mentions. Blair-Goldensohn, et al. found the system to be highly precise for queried online services such as restaurants or hotels. In addition, the system was general enough to produce quality sentiment analysis for all online service types [24].

3.3 Nowcasting

The related works include the following four categories: time series with Google Trends, algorithmic models, early detection of influenza, and lightweight methods to estimate influenza rates.

3.3.1 Time Series with Google Trends

Choi, et al. [1] familiarizes readers with Google Trends data and illustrates some simple forecasting methods that use Google Trends for short-term economic predictions (i.e., predicting the present, contemporaneous forecasting, or nowcasting for economic reports) [1]. Google Trends is a real-time daily and weekly index of the volume of queries that users enter into Google. In short, Choi, et al. combines Google Trends data with AR(1) time series models to generate a 4-6 week forecasting lead. The paper demonstrates AR(1) models for the U.S. Census Bureau Advance Monthly Sales for Retail and Food Services report; the US Department of Labor Initial claims for unemployment benefits report; the Hong Kong Tourism Board’s monthly visitor arrival
statistics report; and the Roy Morgan Consumer Confidence Index for Australia report. Finally, Choi, et al. found that simple seasonal AR time series models that included relevant Google Trends variables tend to outperform models that excluded the predictors by 5% to 20% [1].

3.3.2 Algorithmic Models

Breiman argues for the statistical community to move away from exclusive dependence on data models

(i.e., response variables = f (predictor variables, random noise, parameters)), and adopt a more diverse set of tools (i.e., algorithmic modeling—black box models) [25]. Breiman suggest using powerful algorithm models such as: neural nets, decision trees, support vector machines, and tree ensemble methods. However, Breiman notes trees are great and get an A+ for interpretability, but only get a B on prediction. Thus, Breiman recommends using random forests or neural nets, which have an A+ for prediction, but an F for interpretability. The recommendations are from Breiman’s work experience and shows many users of forecast applications prefer prediction over interpretation. Thus, the goal of a model is not interpretability, but accurate information. Furthermore through three examples, Breiman demonstrates how higher predictive accuracy is associated with more reliable information about the underlying model relationships, and algorithmic models can give better predictive accuracy than data models [25].
3.3.3 Early Detection of Influenza

Underlying models and causes of seasonal influenza are debated but include: seasonal host health, crowding, ambient temperature, indoor heating, air travel, bulk aerosol transport (i.e., coughing and sneezing), and El Nino [26]. In addition, seasonal influenza causes tens of millions of respiratory illnesses and 250,000 to 500,000 deaths worldwide each year [2]. Because existing surveillance networks have a 1-2 week reporting lag, Ginsberg, et al. [2] present an automated early detection method for disease activity. Ultimately, the paper uses a correlation model between web search queries and influenza like illness (ILI) reports from the Center for Disease Control (CDC). Ginsberg, et al. does not propose replacement of traditional surveillance networks or laboratory-based diagnoses surveillance, but sees the early detection as a way to help identify a need for public health inquiry with traditional surveillance to identify the pathogens involved; thus, helping the public health officials to mount a more effective early response to the influenza outbreak. Ginsberg, et al. found the automated system could estimate influenza outbreaks daily, whereas, traditional system estimates required 1-2 weeks to gather and process the surveillance data [2].

3.3.4 Lightweight Methods to Estimate Influenza Rates and Alcohol Sales Volume from Twitter Messages

Culotta’s research shows that tracking a small number of keywords allows influenza rates and alcohol sales volume to be estimated with a high rate of accuracy [27]. The approach was validated with 570 million Twitter messages collected over an eight
month period and data from the U.S. Centers for Disease Control (CDC), and alcohol sales reported by the U.S. Census Bureau. In short, Culotta’s influenza approach first uses a bag-of-words document classifier with logistic regression to predict whether a Twitter message is actually reporting an ILI symptom; thus, reducing the data noise and twitter data set before generating predictions. The classified tweets are then used in the Ginsburg et al. logistic regression approach to predict the CDC’s ILI report values. As a result, Culotta’s approach reduces the volatility, from data noise, to more precisely test for their hypothesis that the content of Twitter messages at time \(i\) correlates with the CDC’s ILI report value at time \(i\) \[27\]. Culotta’s filtering approach looks similar to approaches used to mine web documents (e.g., web pages). However, Culotta takes the traditional document relevance, ranking, and a classification technique used on information retrieval (IR) and applies the techniques to tweets.
CHAPTER 4. SOCIAL MEDIA EVENT DETECTION

Chapter 4 describes the K-SPRE methodology, the social media event detection (SMED) algorithm, and the parallel social media event detection (PSMED) algorithm.

4.1 Social Media Event Detection

In short, social media event detection finds user specified events within the social media.

4.1.1 K-SPRE

K-SPRE \textit{(k-spree)} is the combination of four methodologies used in both the SMED and PSMED algorithms: (1) k-nearest-neighbor [28], (2) social media spatial relations, (3) probabilistic soft logic [22], and (4) referenced events. K-SPRE’s methods are used by SMED and PSMED to find a user specified event (i.e., referenced event) within social media’s spatial relations discovered by probabilistic soft logic and k-nearest-neighbor. K-SPRE uses two social media datasets, (1) a social media data set with trigger-events, and (2) a social media data set with a final-event. And for each instance in the social media final-event dataset, we use the geospatial KNN methodology to search for the k nearest social media instances in the trigger-event dataset. Next, PSL rules between the k nearest social media trigger-events and final-event instances are calculated and written to a file. Further post processing of the K-SPRE output is used to
find the frequency of the event type (i.e., flu event) which satisfies a given threshold rule weight (e.g., RW \geq 0.70).

First, the K-SPRE methodology works as follows: given \( k = 2 \) neighbors and a test point ‘\( x \)’, we compute the test point’s proximity by calculating the correlation, Euclidean distance, or geospatial distance between the point ‘\( x \)’ and the rest of the data points in the training set. Figure 4.1 shows an example with a test point ‘\( x \)’ with the training points represented as squares and filled circles.

In this example, the \( k = 2 \) nearest neighbors of test point ‘\( x \)’ refers to finding the two neighbors that are closest to ‘\( x \)’ using Euclidean distance. The data point ‘\( x \)’ has two found neighbors, a square and a filled circle. Thus, in Figure 4.1 a PSL rule is built between the ‘\( x \)’ point (i.e., final-event), the square (i.e., 1st trigger-event), and the filled circle (i.e., 2nd trigger-event) because they are the two closest neighbors to ‘\( x \)’.

![Figure 4.1 Plot of K-SPRE, where \( k = 2 \)
Second, a PSL rule is generation from the final-event and the trigger-events. Table 4.1 and Figure 4.2 illustrate a simple example with fictitious values. Step (1), the distances between the social media final event (i.e., social media with flu in the content) and the social media trigger events (i.e., social media with fever or nausea in the content) are calculated. Note, in a real world example the distance should be calculated with Spherical Law of Cosines or a similar formula. However, for this simple example the distance between two social media points will be calculated with

\[ d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \]

Figure 4.2 shows the three distances for the PSL rule between points A, B, and C. The three distances are as follows: distance AB = 1.4, distance AC = 1.4, and distance BC = 2. Step (2), the weights from the instance type frequency for each individual point are calculated by the formula

\[ w = \frac{\text{instance type frequency}}{\text{total of all frequencies}} \]

For example, the weights for the points in Figure 4.2 are as follows: point A weight is \( \frac{3}{3+15+9} = 0.11 \), point B weight is \( \frac{15}{3+15+9} = 0.56 \), and point C weight is \( \frac{9}{3+15+9} = 0.33 \). And finally, the rule weight is calculated from the average distance, which is normalized from 0.0 to 1.0 using the maximum distance. The formula for rule weight is

\[ RW = \frac{\text{Distance}_1 + \cdots + \text{Distance}_n}{n} \cdot \frac{1}{\text{Max}(\text{Distance}_1, \ldots, \text{Distance}_n)} \]

where \( n \) is the number of distances in the rule. The rule weight for this example is \( RW = \frac{1.4 + 1.4 + 2.0}{3} \cdot \frac{1}{2.0} = 0.80 \) [22]. Thus, the generalized formula

\[ \text{RW: trigger\_event}_1(w_1) \land \text{trigger\_event}_2(w_2) \Rightarrow \text{final\_event}(w_3) \text{ becomes} \]

\[ 0.80: \text{nausea}(0.33) \land \text{fever}(0.56) \Rightarrow \text{flu}(0.11). \]

Further post processing of all the
generated PSL rules is used to find the frequency of the event type, for this example a flu event, which satisfies the threshold rule weight $RW \geq 0.70$.

Table 4.1 Social media instances

<table>
<thead>
<tr>
<th>Points</th>
<th>Instance Type</th>
<th>Instance Type Frequency</th>
<th>Lat</th>
<th>Long</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>final event = flu</td>
<td>3</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>B</td>
<td>trigger event = fever</td>
<td>15</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>C</td>
<td>trigger event = nausea</td>
<td>9</td>
<td>6</td>
<td>5</td>
</tr>
</tbody>
</table>

Figure 4.2 Plot of social media instances
4.1.2 Social Media Event Detection (SMED) Algorithm

Algorithm 1 Social Media Event Detection (SMED)
1: Let $k$ be the number of nearest neighbors and $D$ be the set of training points
2: FOR each test point $z = (x', y')$
3: DO
4: Compute $d(x', x)$, the distance between $z$ and every point $(x, y) \in D$
5: Select $D_z \subseteq D$, the set of $k$ closest training points to $z$
6: $\text{pslRule} \leftarrow \text{build_psl_rule}(z, D_z)$
7: $\text{pslRules} \leftarrow \text{add}(\text{pslRules}, \text{pslRule})$
8: OD
9: Return($\text{pslRules}$)

Function to build PSL rule

Variables
F: social media final event, $z$
T: social media trigger event, $D_z[i]$
S: a set of social media content
L: a set of locations
D: a set of timestamps
$\text{t}_f$: a set of social media instance types ‘t’ and frequencies ‘f’
$T_F$: a set of distances between a social media trigger event ‘T’ and a social media final event F
$T_T$: a set of distances between two social media trigger events ‘T’
n: number of elements
RW: a rule weight
W: a set of social media trigger and final event types with weights

1: Function $\text{build_psl_rule}(z, D_z)$
2: $S \leftarrow \text{add}(S, z.$instance.$social.media.content)$
3: $L \leftarrow \text{add}(L, z.$instance.$location)$
4: $D \leftarrow \text{add}(D, z.$instance.$timestamp)$
5: $t_f \leftarrow \text{add}(t_f, z.$instance.$type.and.frequency)$
6: FOR each instance of $D_z$
7: DO
8: $S \leftarrow \text{add}(S, D_z[i].$instance.$social.media.content)$
9: $L \leftarrow \text{add}(L, D_z[i].$instance.$location)$
10: $D \leftarrow \text{add}(D, D_z[i].$instance.$timestamp)$
11: $t_f \leftarrow \text{add}(t_f, D_z[i].$instance.$type.and.frequency)$
12: $T_F \leftarrow \text{add}(T_F, \text{distance}(z, D_z[i]))$
13: $T_T \leftarrow \text{add}(T_T, \text{distance}(D_z[i], D_z))$
14: OD
15: n ← sum(TF.length, TT.length)
16: max_distance ← get_max_distance(TF, TT)
17: total_distance ← sum_distances(TF, TT)
18: $RW ← \frac{\text{total_distance}}{n} + \frac{1}{\text{max.distance}}$
19: total_frequency ← sum_frequencies(tf)
20: FOR each tf
21: DO
22: W ← concatenate(type, $\frac{\text{frequency}}{\text{total.frequency}}$)
23: OD
24: rule ← concatenate(RW, W, S, L, D)
25: RETURN rule

4.2 Social Media Event Detection Method on a Cloud Computing Environment

This section first introduces MapReduce and Spatial Hadoop for the cloud computing environment, and concludes with the PSMED algorithm.

4.2.1 MapReduce and SpatialHadoop

MapReduce is a programming model (i.e., software framework) which supports parallel and distributed computing on massive data sets over a cluster of commodity servers [29, 30]. Google originally developed MapReduce using established principals in parallel and distributed processing from over several decades [30]. MapReduce utilizes the popular divide and conquer approach for handling large data problems. For example, the framework partitions a large problem into smaller independent sub problems that can be handled in parallel by many machines in a cluster. Then the intermediate results from each individual sub problem are combined into a final output [30].
programming overhead, the framework hides the implementation of the following data
flow steps: data partitioning, mapping, synchronization, communication, and scheduling.
Yet, there are two user interface functions, the map function and the reduce function,
which can be overwritten to achieve specific functionality [29]. The major open source
commercial MapReduce implementations are the Apache Hadoop and the Microsoft
Dryad [29].

SpatialHadoop is an extended MapReduce library for spatial data analysis, but
more importantly, SpatialHadoop introduces standard spatial indexes and MapReduce
components that allow developers, practitioners, and researchers to design and implement
new spatial operations efficiently [31-34]. By contrast, traditional MapReduce data
analysis tools are not flexible. For instance, industry tools (e.g. ESRI suite of GIS tools)
and academic tools (e.g. Parallel-Secondo, MD-HBase, and Hadoop-GIS) lack
integration with the MapReduce library core—Hadoop core. Hence, these industry and
academic tools are limited by their built in functions and indexes [31]. By contrast,
Hadoop core is a framework designed to efficiently process massive amounts of data in a
distributed fashion [31, 32]. SpatialHadoop solves this limitation by building its
functionality into the core of Hadoop. Therefore, SpatialHadoop operations, analysis
techniques, awareness of spatial data, and new second party spatial constructs are run
inside the Hadoop core making them more efficient with query processing [31, 33].
Furthermore, SpatialHadoop can be used to build scalable applications for massive spatial
datasets. MNTG, SHAHED, and TAREEG are three examples of SpatialHadoop
applications. MNTG is a web based traffic generator developed on the real road global
network. SHAHED is a spatial data analysis tool for exploring NASA’s 500TB archive
containing remote sensing data. And, TAREEG is a web service that extracts real spatial datasets from OpenStreetMap [31, 35-37]. Moreover, future work has been proposed to extend the core of SpatialHadoop with temporal support (e.g. Spatio-temporal Hadoop) [31].

4.2.2 Parallel Social Media Event Detection (PSMED) Algorithm

This work proposes the parallel social media event detection (PSMED) algorithm (i.e., a parallel version of the SMED algorithm), which detects user specified social media events on a cloud computing environment. This section describes the main steps of the algorithm and the pseudo code. The PSMED algorithm has two main phases. As an overview of the first phase, the mapper method reads a cache of social media final event instances F, and the social media trigger event instances T into each node. Next, the mapper calculates the distance between each social media final event F in the cache and the social media trigger event T instance. And lastly, the social media final event F instance, and distance, is combined with the social media trigger event T instance, and is then emitted with a new key. The new key is used to identify the social media final event F instance. The second phase begins with the reducer finding the ‘k’ nearest distances for each new key (i.e., finding the ‘k’ nearest distances to the final event F instance). Next, the social media final event instance F, and the ‘k’ nearest social media trigger event instances T are used to calculate a probabilistic soft logic (PSL) rule [22]. Finally, the reducer then emits the PSL rule, social media content, location(latitude, longitude), and date for each social media final event and it’s trigger events.
The PSL rule function in the reducer class utilizes data from the social media final event instance F and the ‘k’ nearest social media trigger event instances T to build the rule weight RW and the social media type & weights W. Once the PSL rule is assembled, the PSL rule is concatenated with the social media content, locations, and dates from all the event instances that were used to build the PSL rule.

**Algorithm 2 Parallel Social Media Event Detection (PSMED)**

**Input**
(1) A trigger event data set of social media (T), where instances have the following attributes: trigger event keyword; location(latitude, longitude); date/time created; and social media content.
(2) A final event data set of social media (F), where instances have the following attributes: final event keyword; location(latitude, longitude); date/time created; and social media content.

**Output**
(1) A set of probabilistic soft logic (PSL) rules with social media content, where a rule instance with two trigger events and one final event has the following attributes: rule weight; trigger event; weight 1; trigger event 2; weight 2; final event; weight 3; social media 1; social media 2; social media 3; location(latitude, longitude) 1; location(latitude, longitude) 2; location(latitude, longitude) 3.

**Variables**
F: set of social media final events
T: social media trigger event

1: **Class Mapper**
2:   Method Setup ()
3:     F ← add( F, final.event.records)
4:     Method Map(key, T)
5:     FOR each instance of F
6:       DO
7:         new_key ← final.event.index
8:         distance ← distance(T, F_i)
9:       T ← concatenate(T, F_i, distance)
10:      Emit(new_key, T)
11:     OD

12: **Class Reducer**
13:   Method Setup()
14:     k ← 2
15: Method Reduce(new_Key, T’s[T₁, T₂, T₃, ...])
16: //retrieve k instances from the
    // T’s[T₁, T₂, T₃, ...] iterator set with the smallest distances
    k.nearest.neighbors ← get_k_nearest_neighbors(T’s[T₁, T₂, T₃, ...])
17: Rule ← build_psl_rule(k.nearest.neighbors)
18: Emit(null, Rule)

Function to build PSL Rule

Variables
F: social media final event
T: social media trigger event
S: a set of social media content
L: a set of locations
D: a set of timestamps
T_F: a set of distances between a social media trigger event ‘T’ and a social media final event F
T_t: a set of social media trigger event ‘T’ locations ’l’
t_f: a set of social media instance types ‘t’ and frequencies ‘f’
T_T: a set of distances between two social media trigger events ‘T’
n: number of elements
RW: a rule weight
W: a set of social media trigger and final event types with weights

1: Function build_psl_rule(k.nearest.neighbors)
2: FOR each instance of k.nearest.neighbors
3:     DO
4:         S ← add(S, instance.social.media.content)
5:         L ← add(L, instance.location)
6:         D ← add(D, instance.timestamp)
7:         T_F ← add(T_F, instance.T.distance)
8:         T_t ← add(T_t, instance.T.location)
9:         t_f ← add(t_f, instance.type.and.frequency)
10:     OD
11: FOR each T_t
12:     DO
13:         T_T ← add(T_T, distance(between.each.element.in.T_t))
14:     OD
15: n ← sum(T_F.length, T_T.length)
16: max_distance ← get_max_distance(T_F, T_T)
17: total_distance ← sum_distances(T_F, T_T)
18: RW ← \frac{\text{total.distance}}{n} \times \frac{1}{\text{max.distance}}
19: total_frequency ← sum_frequencies(t_f)
20: FOR each t_f
21: DO
22: \[ W \leftarrow \text{concatenate(type, } \frac{\text{frequency}}{\text{total frequency}} \text{)} \]
23: OD
24: rule \leftarrow \text{concatenate(RW, W, S, L, D)}
25: RETURN rule
CHAPTER 5. SENTIMENT ANALYSIS

Chapter 5 presents the sentiment analysis method and algorithm used in this work.

5.1 Sentiment Analysis Method

This work uses sentiment analysis to index the corpus of each social media instance within the user specified event. This derived feature is a measurement of the public’s sentiment (i.e., strength or weakness) about the specified event and is used as an explanatory variable in the nowcast model: index 1 for positive sentiment, index 0 for neutral sentiment, and index -1 for negative sentiment. Furthermore, the indexes are aggregated and normalized for a specified time window (e.g., day, week, month) to be used as a derived attribute for the nowcast model.

5.2 Sentiment Analysis Algorithm

Algorithm 3 Sentiment Analysis
1: pos.words ← read list of positive words
2: neg.words ← read list of negative words
3: tweet ← read tweet
4: tweet ← clean(tweet)
5: tweet ← toLower(tweet)
6: words ← unList(tweet)
7: FOR each words
8: DO
9:   pos.matches ← match(words[i], pos.words)
10:    neg.matches ← match(words[i], neg.words)
11:    OD
12:    score ← sum(pos.matches) - sum(neg.matches)
13:    Return(score)
CHAPTER 6. NOWCASTING MODEL

Chapter 6 presents the methodologies used to create the nowcasting model.

6.1 Nowcasting Model

6.1.1 Artificial Neural Network (ANN)

The first applications of artificial neural networks (ANN) were developed by Widro and Hoff in the 1960’s [38]. Since the 1960’s, ANNs have been used in such applications as telecommunications, air-conditioning systems for automotive systems, laser controls, machine-printing character recognition, quality control in manufacturing, petroleum exploration, and to spot cancerous cells in medicine [38]. Thus, ANN’s successful track record, diversity, and flexibility made it an ideal application for regression models. ANN’s benefits include being free from statistical assumptions, and it is also a robust application when used with missing and inaccurate data sets. As one would guess, there are many variations on the ANN because of its long history and application diversity. However, one type of ANN dominates the field. The back propagated multilayer artificial neural network is estimated to be used in 80% of all applications and is the most widely used in time series forecasting [39]. Thus, this work
will use the back propagated multilayer artificial neural network to build the nowcasting model.

The methodology of an artificial neural network is based on the complex learning system of the human brain (i.e., a closely interconnected set of neurons). However, ANNs can only imitate a basic level of learning. For a simple example, as shown in Figure 6.1, the inputs \( X_i \) (i.e., the data set) are weighted \( W_i \), collected, and combined into a linear combination function \( \Sigma \), which is then put into an activation function to reduce the linear combination value into a range (e.g., three common range types are hard limiters \([-1 \text{ or } 1]\), threshold logic elements \([0 \text{ to } 1]\), and sigmoidal nonlinearities \([-1 \text{ to } 1]\)) [20, 40]. The range value is then used as the output response \( Y \). Furthermore, the output response \( Y \) can be utilized as input for additional hidden layers to build more complex learning systems. Moreover, ANNs use supervised learning with back propagation to train the regression model. For example, when the error between the known report value (\( Y \)) and output response \( Y \) from the activation function is too large, back propagation is used to incrementally adjust the weights in the linear combination to generate a new output response \( Y \) [40].

![Figure 6.1 Artificial neural network with back propagation](image-url)
6.2 Evaluation Methods

This work uses the Markov first-order autoregressive scheme AR(1) as a comparison model to the nowcast model, which are then evaluated by using five popular statistical measures.

6.2.1 Markov First-Order Autoregressive Scheme Methodology

Markov first-order autoregressive scheme (i.e., AR(1)) is a traditional time series methodology of the form

$$U_t = pU_{t-1} + V_t \quad -1 \leq p \leq 1$$

where the value $U_t$ at time $t$ depends on its value in time period $t - 1$, a random term $V_t$, and $p$ (rho) the coefficient of autocorrelation that lies between $-1$ & $1$. There are several ways to estimate $p$: Durbin-Watson d statistic, first difference method, ordinary least squares residual, and Cochrane-Orcutt two-step method [18]. Furthermore, the word autoregressive is reference to the regression of $U_t$ on itself lagged by one period. In addition, the $U_t$ equation is first-order because $U_t$ and its immediate past value are correlated (i.e., the maximum lag is one time period) [18]. This work will use the Markov first-order autoregressive scheme (i.e., AR(1)) as a base line comparison against the nowcasting model.

6.2.2 Evaluation Methodologies

This work will use five popular measures for evaluating numeric prediction of the artificial neural network regression model (i.e., nowcasting model) and the Markov first-
order autoregressive model (i.e., traditional time series model AR(1)). The five popular measures are (1) root mean squared error, (2) mean absolute error, (3) root relative squared error, (4) relative absolute error and (5) correlation coefficient [41].

(1) Root mean squared error: $\sqrt{\frac{(p_1 - a_1)^2 + \cdots + (p_n - a_n)^2}{n}}$

(2) Mean absolute error: $\frac{|p_1 - a_1| + \cdots + |p_n - a_n|}{n}$

(3) Root relative squared error: $\sqrt{\frac{(p_1 - a_1)^2 + \cdots + (p_n - a_n)^2}{(a_1 - a_{\text{Mean}})^2 + \cdots + (a_n - a_{\text{Mean}})^2}}$

(4) Relative absolute error: $\frac{|p_1 - a_1| + \cdots + |p_n - a_n|}{|a_1 - a_{\text{Mean}}| + \cdots + |a_n - a_{\text{Mean}}|}$

(5) Correlation coefficient: $\frac{S_{PA}}{\sqrt{S_p S_A}}$, where

$S_{PA} = \frac{\Sigma_i (p_i - p_{\text{Mean}})(a_i - a_{\text{Mean}})}{n - 1}$, and

$S_p = \frac{\Sigma_i (p_i - p_{\text{Mean}})^2}{n - 1}$, $S_A = \frac{\Sigma_i (a_i - a_{\text{Mean}})^2}{n - 1}$
Chapter 7 presents a framework for nowcasting (i.e., predicting the present) with social media, and a brief description of a nowcasting case study with Twitter and CDC data.

7.1 Case Study Description

This case study utilizes the presented framework for nowcasting with social media to nowcast the Center for Disease Control (CDC) influenza like illness (ILI) reports, utilizing Twitter as the social media data source.

7.2 General KDD Framework

This work will follow the KDD methodology for combining nowcasting and social media. Knowledge Discovery from Data (KDD) refers to a set of activities designed to extract new knowledge from complex data sets [42]. Figure 7.1 shows a condensed diagram of the KDD activities [43], and in general, KDD has 8 steps:

step 1, application domain understanding—fully understand the problem that is being solved; step 2, data selection—selecting a data set by focusing on a subset of variables, or data sample; step 3, data preprocessing—which includes: cleaning and integration, reduction, transformation and discretization [28]; step 4, exploratory analysis—
preliminary investigation (e.g., summary statistics and visualization) of the data in order to better understand the data’s characteristics [28]; step 5, match goal to data mining tasks—match the goal to one of the following methods: classification, association, clustering, or deviation detection (i.e., prediction); step 6, data mining; step 7, interpret data mined results—evaluation and visualization of models and data; step 8, action on discovered data—incorporate new knowledge into another system.

Figure 7.1 Activities diagram of Knowledge Discovery from Data (KDD) [43]

7.3 Framework for Nowcasting with Case Study

The framework for nowcasting with social media includes six main phases. Four phases are illustrated in Figure 7.2, while Figure 7.3 illustrates the last two phases. Each phase is a category of KDD’s activity diagram in Figure 7.1. However, this work will use
slightly different descriptive names for some of the phases. For example, phase 1, the data selection phase is the same as KDD’s data selection step; phase 2, the cleaning & integration phase is part of KDD’s data preprocessing step; phase 3, the reduction phase is an additional part of KDD’s data preprocessing step; phase 4, the transformation & discretization phase is the final part of KDD’s data preprocessing step; phase 5, the regression phase is the same as KDD’s data mining step; and phase 6, the evaluation phase is the same as KDD’s interpret data mined results step. Each phase of the framework has several sub steps that will be reviewed later in this chapter.

Figure 7.2 Nowcasting with social media—first four framework phases
7.3.1 Data Selection Phase 1

The data selection phase is the first phase of nowcasting with social media. This phase is used to select social media with content that can be used as causal variables in the nowcast model to predict a report value (Y). But what is social media? Social media is understood to be data that individuals create, voluntarily share, and contains a minimum of three key features: (1) shared media (e.g., text); (2) date/time stamp (e.g., 12-24-15 1:22:43); and (3) location (Lat, Long) (e.g., -85.108649, 41.114951). Furthermore, major social media generators include Facebook, Twitter, Google, Yahoo, and Instagram. However, social media’s attributes and content are not homogenous
between data generators. Thus, social data sources should be selected to satisfy causal model variables needed for the regression model in phase 5 of Figure 7.3. For example, this work has selected Twitter as a data source. The three main reasons for this choice were: first, the ability to query for causal variables using keywords; second, a date and time stamp attribute per instance; and third, a possible location attribute per instance. Yet, the raw Twitter data uses a JavaScript Object Notation (JSON) format with many attribute-value pairs that are not needed, and will need to be preprocessed. Figure 7.4 shows a single raw Twitter instance.

```
{"created_at":"Tue Jan 05 06:31:53 +0000 2016","entities":{"hashtags":[]},"symbols":[],"urls":[],"user_mentions":[]},"geo":true,"id":684261015425519600,"id_str":684261015425519616,"keywords":["beautiful","familyØY"πi"pictures","edition","meet","cousin ","boyfriend","can’t","wait"],"lang":"en","location":{"lat":41.395049,"lng":-81.54996},"retweet_count":0,"retweeted":false,"screen_name_lower":"_lyknootha","text":"My cousin and her boyfriend took some beautiful pictures!! I can’t wait to meet our new edition to the familyØYπi"timestamp":1451975513000,"type":"tw","user":{"followers_count":196,"id":3015630268,"location":null,"screen_name":"_LykNo Otha","verified":false}}
```

Figure 7.4 Single raw Twitter instance (i.e., Tweet)

In addition to selecting a social media data source, the nowcasted report’s previous values (Y) are needed for the regression phase, as seen in Figure 7.3. This work will build a nowcasting model to predict the CDC’s influenza report; thus, we will utilize the CDC’s historical values for the percent weighted influenza like illness attribute (i.e., % WEIGHTED ILI). However, the raw CDC data set, as shown in Table 7.1, contains many years of data and attributes that are not needed for this work, and will need to be preprocessed.
Table 7.1 Sample (a) and (b) of the CDC's data set attributes for influenza like illness (ILI)

(a)

<table>
<thead>
<tr>
<th>REGION TYPE</th>
<th>REGION</th>
<th>YEAR</th>
<th>WEEK</th>
<th>%UNWEIGHTED ILI</th>
<th>AGE 0-4</th>
<th>AGE 25-49</th>
</tr>
</thead>
<tbody>
<tr>
<td>National</td>
<td>X</td>
<td>2015</td>
<td>48</td>
<td>1.74463</td>
<td>4428</td>
<td>3004</td>
</tr>
<tr>
<td>National</td>
<td>X</td>
<td>2015</td>
<td>49</td>
<td>1.83159</td>
<td>4553</td>
<td>2894</td>
</tr>
<tr>
<td>National</td>
<td>X</td>
<td>2015</td>
<td>50</td>
<td>1.98367</td>
<td>5078</td>
<td>2959</td>
</tr>
<tr>
<td>National</td>
<td>X</td>
<td>2015</td>
<td>51</td>
<td>2.43396</td>
<td>5337</td>
<td>2896</td>
</tr>
<tr>
<td>National</td>
<td>X</td>
<td>2015</td>
<td>52</td>
<td>2.42425</td>
<td>5652</td>
<td>3263</td>
</tr>
<tr>
<td>National</td>
<td>X</td>
<td>2016</td>
<td>1</td>
<td>2.00931</td>
<td>4730</td>
<td>3595</td>
</tr>
<tr>
<td>National</td>
<td>X</td>
<td>2016</td>
<td>2</td>
<td>2.00775</td>
<td>4523</td>
<td>3162</td>
</tr>
<tr>
<td>National</td>
<td>X</td>
<td>2016</td>
<td>3</td>
<td>2.12254</td>
<td>4731</td>
<td>3205</td>
</tr>
</tbody>
</table>

(b)

<table>
<thead>
<tr>
<th>AGE 25-64</th>
<th>AGE 5-24</th>
<th>AGE 50-64</th>
<th>AGE 65</th>
<th>ILITOTAL</th>
<th>NUM. OF PROVIDERS</th>
<th>TOTAL PATIENTS</th>
<th>% WEIGHTED ILI</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>5040</td>
<td>1237</td>
<td>819</td>
<td>14528</td>
<td>2034</td>
<td>832729</td>
<td>1.7336</td>
</tr>
<tr>
<td>X</td>
<td>5314</td>
<td>1251</td>
<td>868</td>
<td>14880</td>
<td>2012</td>
<td>812410</td>
<td>1.8488</td>
</tr>
<tr>
<td>X</td>
<td>5070</td>
<td>1253</td>
<td>935</td>
<td>15295</td>
<td>1983</td>
<td>771045</td>
<td>1.9454</td>
</tr>
<tr>
<td>X</td>
<td>4284</td>
<td>1255</td>
<td>998</td>
<td>14770</td>
<td>1970</td>
<td>606829</td>
<td>2.4362</td>
</tr>
<tr>
<td>X</td>
<td>3966</td>
<td>1519</td>
<td>1172</td>
<td>15572</td>
<td>1938</td>
<td>642342</td>
<td>2.5209</td>
</tr>
<tr>
<td>X</td>
<td>4453</td>
<td>1639</td>
<td>1155</td>
<td>15572</td>
<td>1990</td>
<td>774991</td>
<td>2.0566</td>
</tr>
<tr>
<td>X</td>
<td>5323</td>
<td>1378</td>
<td>1010</td>
<td>15396</td>
<td>1991</td>
<td>766828</td>
<td>2.0971</td>
</tr>
<tr>
<td>X</td>
<td>5307</td>
<td>1293</td>
<td>944</td>
<td>15480</td>
<td>1983</td>
<td>729314</td>
<td>2.1950</td>
</tr>
</tbody>
</table>

7.3.2 Cleaning & Integration Phase 2

The second phase of nowcasting with social media is the cleaning & integration phase, as shown in Figure 7.2. Phase 2 contains three main sub sections: (1) query data for causal model variables; (2) clean data; and (3) clean report values (Y). Essentially, this phase queries the social media data for causal model variables (e.g., query flu, fever, and nausea as causal variables for modeling the CDC’s ILI report value (Y)).

Furthermore, querying a data set for causal model variables can be defined as: given a query function for social media $Q(\cdot)$, a set of causal variables $\{x_1, x_2, \ldots, x_n\}$, and a set of social media data $S$, then the queried data set with causal model variables is $D$,

$$D = \{Q(x_1), Q(x_2), \ldots, Q(x_n)\} \subseteq S.$$  Additionally, this phase cleans the data by
removing all unwanted attributes and instances from the queried data and report values (Y).

This work removed attributes that were not needed, and instances that were missing attribute information. What's more, the keyword queried (e.g., flu, fever, or nausea) along with the query frequency were added to the instances, and duplicate time/location instances between the trigger events and final event were removed. In addition, the social media’s text attribute instances were cleaned of all unwanted punctuation and text (e.g., https, emojis, and commas) and written to a comma separated file for use in phase 3. For this work, Table 7.2, 7.3, and 7.4 illustrate a sample of the cleaned Tweets, cleaned Tweets keyword frequency, and cleaned CDC data.

Table 7.2 Sample of queried and cleaned Twitter data

<table>
<thead>
<tr>
<th>week</th>
<th>long</th>
<th>lat</th>
<th>created</th>
<th>keyWord</th>
<th>text</th>
</tr>
</thead>
<tbody>
<tr>
<td>week 1</td>
<td>-121.0544444</td>
<td>39.242222</td>
<td>16:50:05</td>
<td>flu</td>
<td>wind mph sse barometer steady temperature f rain today humidity</td>
</tr>
<tr>
<td>week 1</td>
<td>-71.281090</td>
<td>41.945080</td>
<td>16:50:05</td>
<td>fever</td>
<td>current weather attleboro cloudy f humidity wind mph visibility mi</td>
</tr>
<tr>
<td>week 1</td>
<td>-122.622917</td>
<td>45.534903</td>
<td>16:50:05</td>
<td>cough</td>
<td>barista us starbucks tocofxxnn hospitality veterans job hiring</td>
</tr>
<tr>
<td>week 2</td>
<td>-122.403853</td>
<td>37.788791</td>
<td>16:50:05</td>
<td>flu</td>
<td>needed coffee starting exciting day capital one caf san francisco ca</td>
</tr>
<tr>
<td>week 2</td>
<td>-122.036350</td>
<td>37.368830</td>
<td>16:50:08</td>
<td>fever</td>
<td>join obre team see latest purchasing job opening</td>
</tr>
<tr>
<td>week 2</td>
<td>-73.687746</td>
<td>40.661776</td>
<td>16:50:08</td>
<td>cough</td>
<td>join oberlender dorfman inc team see latest accounting job</td>
</tr>
<tr>
<td>week 2</td>
<td>-77.171091</td>
<td>38.882334</td>
<td>16:50:12</td>
<td>fatigue</td>
<td>want work cricket wireless were hiring fallschurch va click details</td>
</tr>
<tr>
<td>week 3</td>
<td>-118.243885</td>
<td>34.052234</td>
<td>16:50:09</td>
<td>flu</td>
<td>this customerservice job might great fit tibetan interpreter</td>
</tr>
<tr>
<td>week 3</td>
<td>-80.233104</td>
<td>26.062866</td>
<td>16:50:12</td>
<td>fever</td>
<td>were hiring dick apply foodservice specialist</td>
</tr>
</tbody>
</table>

Table 7.3 Sample of cleaned Twitter data, keyword frequency

<table>
<thead>
<tr>
<th>week</th>
<th>keyWord</th>
<th>keyword frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>week 1</td>
<td>flu</td>
<td>47585</td>
</tr>
<tr>
<td>week 1</td>
<td>fever</td>
<td>30588</td>
</tr>
<tr>
<td>week 1</td>
<td>cough</td>
<td>20095</td>
</tr>
<tr>
<td>week 2</td>
<td>flu</td>
<td>7586</td>
</tr>
<tr>
<td>week 2</td>
<td>fever</td>
<td>8587</td>
</tr>
<tr>
<td>week 2</td>
<td>cough</td>
<td>5555</td>
</tr>
<tr>
<td>week 2</td>
<td>fatigue</td>
<td>1585</td>
</tr>
<tr>
<td>week 3</td>
<td>flu</td>
<td>30583</td>
</tr>
<tr>
<td>week 3</td>
<td>fever</td>
<td>25525</td>
</tr>
</tbody>
</table>
The third phase of nowcasting with social media is the reduction phase, as shown in Figure 7.2. This phase contains one main sub section: apply Social Media Event Detection (SMED). In short, SMED utilizes K-SPRE methodology to find a user specified event type within social media. Thus, reducing unrelated social media instances (i.e., data noise) and incomplete instances. The output from SMED can be seen in Table 7.5. Further post processing of SMED’s output is used to find the count of rule instances which satisfies a given threshold (e.g., \( \text{RW} \geq 0.70 \)). This count of rule instances is then used as the event type’s frequency. In this case, the event type is flu (i.e., the final event), and the flu event frequency will be used as an attribute in the nowcast model.
Table 7.5 Sample (a) and (b) output attributes from Social Media Event Detection (SMED)

(a)

<table>
<thead>
<tr>
<th>week</th>
<th>Rule ID</th>
<th>trigger_event_1</th>
<th>W1</th>
<th>trigger_event_2</th>
<th>W2</th>
<th>final_event</th>
<th>W3</th>
</tr>
</thead>
<tbody>
<tr>
<td>week 1</td>
<td></td>
<td>bodyAches</td>
<td>0.214256</td>
<td>runnyNose</td>
<td>0.316861</td>
<td>flu</td>
<td>0.468882</td>
</tr>
<tr>
<td>week 1</td>
<td>2</td>
<td>bodyAches</td>
<td>0.196987</td>
<td>chills</td>
<td>0.371923</td>
<td>flu</td>
<td>0.43109</td>
</tr>
<tr>
<td>week 1</td>
<td>3</td>
<td>bodyAches</td>
<td>0.214349</td>
<td>stuffyNose</td>
<td>0.316565</td>
<td>flu</td>
<td>0.469086</td>
</tr>
<tr>
<td>week 2</td>
<td>4</td>
<td>bodyAches</td>
<td>0.213449</td>
<td>stuffyNose</td>
<td>0.316565</td>
<td>flu</td>
<td>0.469086</td>
</tr>
<tr>
<td>week 2</td>
<td>5</td>
<td>bodyAches</td>
<td>0.196214</td>
<td>fever</td>
<td>0.374389</td>
<td>flu</td>
<td>0.429398</td>
</tr>
<tr>
<td>week 2</td>
<td>7</td>
<td>bodyAches</td>
<td>0.208194</td>
<td>soreThroat</td>
<td>0.33619</td>
<td>flu</td>
<td>0.455616</td>
</tr>
<tr>
<td>week 3</td>
<td>8</td>
<td>bodyAches</td>
<td>0.196214</td>
<td>fever</td>
<td>0.374389</td>
<td>flu</td>
<td>0.429398</td>
</tr>
<tr>
<td>week 3</td>
<td>9</td>
<td>bodyAches</td>
<td>0.196987</td>
<td>chills</td>
<td>0.371923</td>
<td>flu</td>
<td>0.43109</td>
</tr>
</tbody>
</table>

(b)

<table>
<thead>
<tr>
<th>week</th>
<th>Rule ID</th>
<th>text_1</th>
<th>text_2</th>
<th>text_3</th>
<th>long_1</th>
<th>lat_1</th>
<th>long_2</th>
<th>lat_2</th>
<th>long_3</th>
<th>lat_3</th>
<th>date_1</th>
<th>date_2</th>
<th>date_3</th>
</tr>
</thead>
<tbody>
<tr>
<td>week 1</td>
<td>1</td>
<td>sale</td>
<td>profe</td>
<td>short</td>
<td>we</td>
<td>bread</td>
<td>-80.799</td>
<td>35.11735</td>
<td>-80.8398</td>
<td>35.2295</td>
<td>-80.9539</td>
<td>35.1498</td>
<td>18:03:42 +00:19:03 +04:16:59:49 +4</td>
</tr>
<tr>
<td>week 1</td>
<td>2</td>
<td>ioux</td>
<td>falls</td>
<td>sattis</td>
<td>tions</td>
<td>-96.733</td>
<td>43.51616</td>
<td>-98.402</td>
<td>40.5974</td>
<td>100.623</td>
<td>43.29887</td>
<td>18:39:59 +02:20:12 +04:16:59:50 +4</td>
<td></td>
</tr>
<tr>
<td>week 1</td>
<td>3</td>
<td>ice</td>
<td>drums</td>
<td>ve</td>
<td>farme</td>
<td>-118.267</td>
<td>34.04146</td>
<td>-118.266</td>
<td>34.0991</td>
<td>-118.249</td>
<td>34.04675</td>
<td>19:06:41 +20:17:01 +04:17:00:11 +4</td>
<td></td>
</tr>
<tr>
<td>week 2</td>
<td>4</td>
<td>ales</td>
<td>profye</td>
<td>short</td>
<td>we</td>
<td>ev</td>
<td>-70.9592</td>
<td>42.03343</td>
<td>-70.6175</td>
<td>41.5523</td>
<td>-70.2382</td>
<td>42.06119</td>
<td>19:47:28 +21:24:55 +04:17:05:04 +4</td>
</tr>
<tr>
<td>week 2</td>
<td>5</td>
<td>sales</td>
<td>profye</td>
<td>short</td>
<td>we</td>
<td>ev</td>
<td>-80.799</td>
<td>35.11735</td>
<td>-80.8398</td>
<td>35.2295</td>
<td>-80.9539</td>
<td>35.1498</td>
<td>18:03:42 +00:19:03 +04:16:54:15 +4</td>
</tr>
<tr>
<td>week 3</td>
<td>7</td>
<td>nca</td>
<td>sient</td>
<td>ed</td>
<td>wo</td>
<td>rywtrdow</td>
<td>77.0375</td>
<td>38.89905</td>
<td>77.0492</td>
<td>38.90444</td>
<td>77.0369</td>
<td>38.90719</td>
<td>18.37:01 +00:42:44 +04:17:04:40 +4</td>
</tr>
<tr>
<td>week 3</td>
<td>8</td>
<td>herset</td>
<td>rt</td>
<td>dippa</td>
<td>fort</td>
<td>tbonco</td>
<td>74.4104</td>
<td>40.64539</td>
<td>-74.4584</td>
<td>40.45977</td>
<td>-74.5991</td>
<td>40.62314</td>
<td>22:02:33 +00:12:50 +04:16:54:19 +4</td>
</tr>
<tr>
<td>week 3</td>
<td>9</td>
<td>king</td>
<td>away</td>
<td>picnic</td>
<td>came</td>
<td>ha</td>
<td>-119.262</td>
<td>36.30079</td>
<td>-119.262</td>
<td>36.30069</td>
<td>-119.262</td>
<td>36.30071</td>
<td>17:06:23 +17:39:46 +04:17:04:05 +4</td>
</tr>
</tbody>
</table>

7.3.4 Transformation & Discretization Phase 4

The fourth phase of nowcasting with social media is the transformation & discretization phase, as shown in Figure 7.2. This phase contains one main sub section: apply sentiment analysis to generate a derived attribute. This phase will use sentiment analysis to create a derived attribute which indicates the strength or weakness of the user’s specified event type during a specific time frame (e.g., day, week, or month) within the social media source, and is then used as a sentiment attribute in the nowcast model. Specifically, the SMED output corpus of each rule instances which satisfies the threshold RW ≥ 0.70 is indexed as 1 for positive sentiment, 0 for neutral sentiment, or -1 for negative sentiment. Furthermore, the sentiment analysis index values are aggregated.
to match the time frame of the report value (Y) (e.g., CDC’s report values are weekly, thus weekly aggregation), and counted to produce a sentiment frequency. A sample output of the sentiment frequency can be seen in Table 7.6.

Table 7.6 Sample output of sentiment analysis

<table>
<thead>
<tr>
<th>week</th>
<th>sentiment frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>week 1</td>
<td>301.00</td>
</tr>
<tr>
<td>week 2</td>
<td>331.00</td>
</tr>
<tr>
<td>week 3</td>
<td>243.00</td>
</tr>
<tr>
<td>week 4</td>
<td>231.00</td>
</tr>
<tr>
<td>week 5</td>
<td>304.00</td>
</tr>
<tr>
<td>week 6</td>
<td>224.00</td>
</tr>
<tr>
<td>week 7</td>
<td>212.00</td>
</tr>
<tr>
<td>week 8</td>
<td>236.00</td>
</tr>
</tbody>
</table>

7.3.5 Regression Phase 5

The fifth phase of nowcasting with social media is the regression phase, as shown in Figure 7.3. Phase 5 contains three main sub sections: (1) split data from phases 2 - 4; (2) apply a machine learning regression method to training set; and (3) apply learned model to test set. This phase splits the data sets from phases 2 - 4 into a training set and a test set. Next, a machine learning regression method is applied to the training set. The training set contains the main attributes: queried keyword frequency, user specified event type frequency, sentiment analysis frequency, and report values (Y). However, the attributes in Table 7.7 have slightly different names: keyword flu frequency, flu event frequency, sentiment frequency, and % weighted ILI. The learned model is then applied to the test set, as shown in Table 7.8. In this case the learned model will be of a regression form. The word regression refers to methods used to predict numeric
quantities [41]. That is, regression is the task of learning a target function $f$ that maps each attribute set $x$ into a continuous-valued output ($Y$) [28]. The result of this phase is a target function (i.e., learned model) that can be used for nowcasting (i.e., a predictive model). Least square linear regression, decision trees (e.g., regression trees), support vector machines (e.g., support vector regression), and artificial neural networks are popular regression methods. In this work, artificial neural networks are used with cross validation to limit overfitting of the training set, while building the learned model. This work’s training data set uses the attributes in Table 7.7, which are averaged and normalized. Furthermore, the CDC’s % weighted ILI report value ($Y$) is normalized during the machine learning regression step. Lastly, the learned regression model is applied to the test set, as seen in Table 7.8, and the test set results (i.e., predictions) are then evaluated in phase 6. In addition, the nowcasting model for each week is built by using the previous week(s) as the training data set, and the current week as the test data set in Table 7.9 and 7.10.

Table 7.7 Sample of the nowcast model's training data set for week 8

<table>
<thead>
<tr>
<th>Week</th>
<th>Keyword Flu Frequency</th>
<th>Flu Event Frequency</th>
<th>Sentiment Frequency</th>
<th>% Weighted ILI</th>
</tr>
</thead>
<tbody>
<tr>
<td>week 1</td>
<td>0.539945</td>
<td>0.570644</td>
<td>0.475393</td>
<td>1.819066</td>
</tr>
<tr>
<td>week 2</td>
<td>0.869224</td>
<td>0.556398</td>
<td>0.642234</td>
<td>1.921719</td>
</tr>
<tr>
<td>week 3</td>
<td>0.801731</td>
<td>0.634460</td>
<td>0.730076</td>
<td>2.363420</td>
</tr>
<tr>
<td>week 4</td>
<td>0.989616</td>
<td>0.658436</td>
<td>0.797959</td>
<td>2.483186</td>
</tr>
<tr>
<td>week 5</td>
<td>1.217197</td>
<td>0.732841</td>
<td>0.892545</td>
<td>2.000557</td>
</tr>
<tr>
<td>week 6</td>
<td>1.420874</td>
<td>0.858430</td>
<td>0.937289</td>
<td>2.061750</td>
</tr>
<tr>
<td>week 7</td>
<td>1.411141</td>
<td>1.000660</td>
<td>0.971204</td>
<td>2.164716</td>
</tr>
</tbody>
</table>

Table 7.8 Sample of the nowcast model's test data set (i.e., prediction) for week 8

<table>
<thead>
<tr>
<th>Week</th>
<th>Keyword Flu Frequency</th>
<th>Flu Event Frequency</th>
<th>Sentiment Frequency</th>
<th>% Weighted ILI</th>
</tr>
</thead>
<tbody>
<tr>
<td>week 8</td>
<td>1.350012</td>
<td>1.029629</td>
<td>1.010984</td>
<td>2.284472</td>
</tr>
</tbody>
</table>
Table 7.9 Sample of the nowcast model's training data set for week 9

<table>
<thead>
<tr>
<th>Week</th>
<th>Keyword Flu Frequency</th>
<th>Flu Event Frequency</th>
<th>Sentiment Frequency</th>
<th>% Weighted ILU</th>
</tr>
</thead>
<tbody>
<tr>
<td>week 1</td>
<td>0.539945</td>
<td>0.570644</td>
<td>0.475393</td>
<td>1.819066</td>
</tr>
<tr>
<td>week 2</td>
<td>0.869224</td>
<td>0.556398</td>
<td>0.642234</td>
<td>1.921719</td>
</tr>
<tr>
<td>week 3</td>
<td>0.801731</td>
<td>0.634460</td>
<td>0.730076</td>
<td>2.363420</td>
</tr>
<tr>
<td>week 4</td>
<td>0.989616</td>
<td>0.658436</td>
<td>0.797959</td>
<td>2.483186</td>
</tr>
<tr>
<td>week 5</td>
<td>1.217197</td>
<td>0.732841</td>
<td>0.892545</td>
<td>2.000557</td>
</tr>
<tr>
<td>week 6</td>
<td>1.420874</td>
<td>0.858430</td>
<td>0.937289</td>
<td>2.061750</td>
</tr>
<tr>
<td>week 7</td>
<td>1.411141</td>
<td>1.000660</td>
<td>0.971204</td>
<td>2.164716</td>
</tr>
<tr>
<td>week 8</td>
<td>1.350012</td>
<td>1.029629</td>
<td>1.010984</td>
<td>2.284472</td>
</tr>
</tbody>
</table>

Table 7.10 Sample of the nowcast model's test data set (i.e., prediction) for week 9

<table>
<thead>
<tr>
<th>Week</th>
<th>Keyword Flu Frequency</th>
<th>Flu Event Frequency</th>
<th>Sentiment Frequency</th>
<th>% Weighted ILU</th>
</tr>
</thead>
<tbody>
<tr>
<td>week 9</td>
<td>1.325026</td>
<td>1.127655</td>
<td>1.071553</td>
<td>2.414823</td>
</tr>
</tbody>
</table>

7.3.6 Evaluation Phase 6

The sixth phase of nowcasting with social media evaluates and compares the results of the learned model. The results (i.e., predictions) are generated by applying the learned model to the test set. Evaluation is a systematic way to evaluate how different learned models work and compare with one another [41]. Five popular measures for evaluation are: (1) root mean squared error, (2) mean absolute error, (3) root relative squared error, (4) relative absolute error, and (5) correlation coefficient. Root mean squared error is the most commonly used evaluation measure. Mean absolute error tends to exaggerate the effects of outliers. Root relative squared error, is the error relative to a simple predictor built from the average actual values of the training data. Relative absolute error is the total error with normalization. And, correlation coefficient measures the statistical correlation between the actual values and the prediction values. For most practical situations, the machine learning regression method with the best numerical
prediction is still the best model no matter which error measure is used [25, 41]. In this work, the previous five evaluation measures will be used to evaluate and compare the learned model to a traditional time series model—the Markov first-order autoregressive scheme (i.e., AR(1)).
CHAPTER 8. EXPERIMENTAL EVALUATION

Chapter 8 presents the experimental setup and evaluation for nowcasting with social media.

8.1 Experimental Setup

8.1.1 Data

This work’s data set will be the popular social media data source Twitter. Twitter is a free major online social media generator and micro-blogging service that asks users ‘what are you doing?’ Users can send and read short 140 character messages called Tweets [7, 9, 10, 12]. Additionally, in 2014, Twitter’s average monthly active users (MAUs) were 255 million, and mobile MAUs reached 198 million with a year-over-year increase of 25% and 31%, respectively [44]. Twitter has several advantages as a data set: (1) it is a good source of real-time data; (2) the data set contains many popular topics, and (3) even though the Twitter API only grants access to a 1% sample, the sample size provides enough data for nowcasting.

Raw Twitter data was collected using Arizona State University’s Tweet Tracker (ASUTT) and the R library (TwitteR) [45, 46]. R’s ability to manipulate matrices without the need for loop iterations, and R’s extended JSON libraries made R an ideal tool for pre
and post processing of the large raw Twitter json files [47]. Tweets were collected from October 2015 to March 2016 within the United States geographical area. In addition, tweets were queried based on 13 individual key words: flu, fever, cough, sore throat, stuffy nose, runny nose, body aches, headaches, chills, fatigue, nausea, vomiting, and diarrhea which resulted in a small dataset of 9.29GB of raw tweets. However, much of the data did not have geo coordinates (i.e., users did not want to share their locations). Consequently, after preprocessing, the size of the Twitter data set was reduced to 1.0GB, or approximately 7.25 million tweets.

8.1.2 Nowcasting and Time Series Model Experimental Setup

The nowcasting and time series models were built using Weka from the University of Waikato. Weka is open source software with a collection of machine learning algorithms for data mining tasks, such as data-preprocessing, classification, regression, clustering, association rules, and visualization [48]. In addition, Weka version 3.7.13 includes a time series forecasting library. Therefore, Weka’s artificial neural network with back propagation was utilized (i.e., a multilayer perceptron) for both the nowcasting model and the AR(1) time series model.

8.1.3 Cloud Computing

The scalable parallel and distributed social media event detection (SMED) algorithm was run on Amazon Web Service (AWS). This work used multiple clusters containing \{1, 2, 4, 8, 16\} nodes. Each node contained an m3.medium instance with each
instance containing a single vCPU, 3.75GB of memory, and 1x4GB SSD storage. Some of the features of the m3.medium instance include high frequency Intel Xeon E5-2670 v2 (Ivy Bridge) processor; SSD-based instance storage for fast I/O performance; and a balance set of resources for computation, memory, and network overhead [49]. The cluster was assembled manually and did not utilize AWS’s software as a service (SaaS) for Hadoop, also known as, Amazon Elastic MapReduce (EMR). Assembling the cluster manually was more cost efficient, and allowed for flexibility to utilize Hadoop and SpatialHadoop open source libraries not available through AWS. The main open source packages utilized on the AWS clusters were Hadoop-2.4.0 [50] and this work’s modified version of SpatialHadoop-2.2.0, which were both used in building the social media event detection (SMED) algorithm [34].

8.2 Experimental Results

8.2.1 Nowcast Model Evaluations

In this work the correlation coefficient is used to measure the statistical correlation between the model’s predicted CDC’s ILI values and the actual CDC’s ILI values. Correlations make it possible to use the value of one variable to predict the value of another. In this case we are looking for the highest correlation coefficient to test for the best performing model (i.e., to test if the predicted CDC’s ILI values correlate with the actual CDC’s ILI values). In this work, as shown in Table 8.1, the nowcast model showed the best performance with a higher correlation coefficient of 0.89 versus the AR(1) model’s correlation coefficient of 0.83. In addition, the nowcast model’s improved
performance is further supported by the nowcast model’s lower statistical error in four measurements: root mean squared error, mean absolute error, root relative squared error, and relative absolute error.

The nowcast model’s improved performance over the AR(1) model is due to advantages and disadvantages of the different data sources used between the two models. For example, the nowcast model has the advantage of using real-time data (e.g., Twitter) where the AR(1) model has the disadvantage of using historical data (e.g., the previous week’s CDC’s ILI report value) when predicting non-trending values. For instance, both the nowcast and AR(1) models’ predictions between 12/19/15 and 02/20/16 look similar in this trending subsection. However, as the predictions continue to 04/16/16 the trend changes, and the AR(1) model’s historical data does not predict the turn in the actual CDC’s ILI report values as well as the nowcast model which uses real-time data. Questions may be raised about the AR(1) model. For example, does an AR(2), AR(3), or AR(n) model have better performance to the AR(1) model? To answer this question, exploratory analysis was used to evaluate the best AR model. The analysis showed the AR(1) model had a higher correlation to the actual CDC’s ILI report values then the AR(2) and AR(3) models. Thus, this work used the AR(1) model as a baseline for comparison to the nowcast model.

The nowcast model showed a 16% to 20%, improvement over the traditional time series model AR(1) in four popular statistical measures, and additionally showed a slightly better correlation coefficient, as shown in Table 8.1. Furthermore, a comparison of the AR(1) model forecasts and the nowcast model forecasts can be seen in Figures 8.1 and 8.2.
Table 8.1 AR(1) and nowcast model statistical error measurements

<table>
<thead>
<tr>
<th>Measurement</th>
<th>AR(1)</th>
<th>Nowcast</th>
<th>Improvement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Root Mean Squared Error</td>
<td>0.38</td>
<td>0.30</td>
<td>20%</td>
</tr>
<tr>
<td>Mean Absolute Error</td>
<td>0.30</td>
<td>0.26</td>
<td>16%</td>
</tr>
<tr>
<td>Root Relative Squared Error</td>
<td>0.76</td>
<td>0.61</td>
<td>20%</td>
</tr>
<tr>
<td>Relative Absolute Error</td>
<td>0.80</td>
<td>0.67</td>
<td>16%</td>
</tr>
<tr>
<td>Correlation Coefficient</td>
<td>0.83</td>
<td>0.89</td>
<td>7%</td>
</tr>
</tbody>
</table>

Figure 8.1 AR(1) model forecasts and CDC’s % weighted influenza like illness values

Figure 8.2 Nowcast model forecasts and CDC's % weighted influenza like illness values
8.2.2 Parallel Social Media Event Detection (PSMED) Evaluation

Admittedly, this work’s 1.0GB of Twitter data is a relatively small data set. However, we were able to improve the social media event detection (SMED) algorithm’s running time by 65%, as shown in Figure 8.3 and 8.4. In short, the improvement was achieved by using Hadoop and modified SpatialHadoop libraries to create the parallel social media event detection (PSMED) algorithm that was distributed over a 16 node Amazon Web Service (AWS) cluster.

Figure 8.3 Normalized time improvement of parallel social media event detection (PSMED) algorithm distributed over an AWS cluster
Figure 8.4 Time improvement of parallel social media event detection (PSMED) algorithm distributed over an AWS cluster
CHAPTER 9. CONCLUSION

Nowcasting events and social media analysis are growing areas of research that have advanced significantly as social media is becoming more popular [4]. Nowcasting with social media creates challenges because of the HACE characteristics of big data (i.e., heterogeneous, autonomous, complex, and evolving associations) [15]. Thus, this thesis proposed a feature extraction method to improve nowcasting with social media as a real-time data source. The proposed social media event detection (SMED) algorithm utilizes K-SPRE methodology to locate data from a user specified event which is then processed with sentiment analysis. In addition, we develop a parallel social media event detection (PSMED) algorithm, which uses the K-SPRE methodology on a cloud environment. PSMED was used to extract features from social media that were used in an adapted artificial neural network to build a nowcasting model. The nowcast model was then used to predict, in real-time, business and government reports for more timely decision making. Furthermore, we complete a case study with real data from Twitter and the Center for Disease Control (CDC) influenza like illness (ILI) reports. In contrast to Culotta’s research [27], our approach utilizes the complete twitter data set when making predictions (i.e., nowcasts). However, the unfiltered data noise added volatility to our model’s predictions. Thus, derived attributes are added to our regression model as a way to reduce the prediction’s volatility. The regression model was inspired by Choi’, et al.
and Ginsberg, et al. previous works, and the derived attributes are built from Santos’, et al. key idea that social events tend to be associated to other spatially and temporally-related nearby activates that can be used to uncover a final event. Utilization of Santos’, et al. key idea produces secondary advantages with further post processing: the approximate geospatial locations of the dynamically changing event over time (i.e., geospatial hotspots containing the flu event per week); popular keyword themes within the geospatial area; and a Tweet corpus with less data noise that can be used in sentiment analysis to determine the strength or weakness of the flu events within the geospatial area. Because of miscommunicated tweets, slang, typos, etc., one cannot say with 100% certainty a tweet belongs to a specific event without interviewing the tweet author. Thus, this work utilizes Santos’, et al. probabilistic soft logic approach to define if tweets are, or are not, from the same event. Additionally, some tweets containing possible flu event content, such as “I don’t feel well” can be missed. Two popular reasons for not discovering tweets with flu events include: (1) Twitter’s API only grants access to a 1% sample of the Twitter data; and (2) about 40% of tweets queried or mined will not include the queried keyword(s) (e.g., don’t feel well). Experiments with predicting the CDC’s ILI report shows nowcasting with social media outperforms the traditional time series AR(1) model by as much as 16% to 20%, in terms of statistical error. Furthermore, by utilizing cloud computing (e.g., Amazon Web Service (AWS) and Hadoop) the PSMED algorithm’s running time was improved by 65%. However, the PSMED algorithm needs to be validated on larger data sets, and different types of social media (e.g., Facebook, Google, Yahoo, and Instagram). In addition, the PSMED algorithm generated a large
overhead relative to the size of the utilized Twitter data set when implemented on Hadoop; as a result, further research is needed for algorithm optimization.
LIST OF REFERENCES
LIST OF REFERENCES


