Can Social Media Support Public Health?
Demonstrating Disease Surveillance using Big Data Analytics

Full paper

Ashwin Kumar T K
Oklahoma State University, USA
tashwin@okstate.edu

Daniel Asamoah
Wright State University, USA
daniel.asamoah@wright.edu

Ramesh Sharda
Oklahoma State University, USA
ramesh.sharda@okstate.edu

Abstract

Rapid growth of the Internet has paved the way for millions of people across the globe to access social media platforms such as Facebook and Twitter. These social media platforms enable people to share information instantaneously. The large volume of information shared on these platforms can be leveraged to identify outbreaks of various epidemics. This will help health professionals to provide timely intervention, which in return could help save lives and millions of dollars. Analysis of information shared on social media is complicated due to its sheer volume, varied formats and velocity of collection. We have addressed this potential problem by making use of a big data analytics platform capable of handling large quantities of streaming data. In this paper we demonstrate how data from social media can be effectively used in the surveillance of disease conditions.

Keywords

Social media, disease surveillance, Internet, information sharing, healthcare analytics

Introduction

The power of electronic social media platforms such as Facebook and Twitter is their ability to connect people across the globe and aid them in sharing information with one another instantaneously. A recent study on social networking statistics revealed that 58% of the world’s population have been on at least one social networking platform; 56% people use Facebook, 14% people use LinkedIn, 11% people use Twitter and 9% use Google+ (Statistic Brain 2014). Given the amount of people on social media platforms sharing huge volumes of data, it is trite to say there is tremendous potential for data analytics to derive deep insights on various topics. Big data technologies are a perfect match for leveraging data from social media because they do not require expensive infrastructure as big data analytics methodology allows commodity hardware to be used to store and analyze the data. In addition to these, big data technologies can handle large quantities of streaming data of varying formats.

Healthcare is one of many areas that can benefit from information posted on these social media platforms. For instance, the use of surveillance methods, enhanced by data analytics and as part of public health initiatives can help discover the emergence of disease outbreaks even before they become full scale.

Information about disease symptoms from various social network platforms can be used to identify and prevent disease outbreak and help health professionals provide timely intervention. Timely detection and intervention of an infectious disease could help save lives and millions of dollars. The Ebola disease, for example, has already accounted for about six thousand deaths (World Health Organisation 2014). One of the challenges to the containment of this disease has been the identification and management of core centers where the disease is mostly located. The ability to do so would help policy makers’ direct valuable resources to those areas in an effort to help contain the disease.
In the past, studies have been done on how location-based social media platforms could be used to as a source of spatio-temporal information. Such social media-derived vital information have been used in applications such as fire-fighting and security (De Longueville et al. 2009).

There are other studies, which analyze social media data for spread of various diseases (Kathy et al. 2013 and Courtney et al. 2010). In Courtney et al.’s (2010) study, data from several blogs and blog posts are analyzed for influenza surveillance. Although temporal analysis of spread of influenza is done in this study, geo-spatial analysis is not feasible because the location of bloggers and people posting replies to blog posts cannot be obtained. A real-time framework for influenza and cancer surveillance is proposed in (Kathy et al. 2013). This framework uses data from Twitter and performs geo-spatial and temporal analysis of data. The data pre-processing step in this framework differentiates actual data from noise based on just the occurrence of keywords like flu and cancer. The keywords can occur in multiple contexts. For example, the following sentences have the same keyword, cancer, but they are not in same context: ("Some people are cancerous to our society"; “I have been diagnosed with blood cancer”). Therefore, it becomes essential to determine if these keywords occur in the right context to be classified as relevant data or noise. We have addressed this problem in this paper by using a NLP tool called TweetNLP (Owoputi et al. 2012), which is used to identify the context of occurrence of keywords. In addition to this, the framework proposed by (Kathy et al. 2013) assumes that the user is always at the location specified in his or her user profile and does not account for their movement. In our case study, we use geo-location attribute from the users’ tweets. These attributes are available if the user shares his or her location with others.

Against this background, we ask, “can big data analytics on social media support disease surveillance?”. In this study, we aim to utilize such location-based information in a healthcare application. We demonstrate the use of Big Data analytics as a health surveillance tool to support public health initiatives targeted at quickly identifying trends of diseases before they escalate into epidemics. Specifically, we identify trends of various disease symptoms using social media data from Twitter and analyze the occurrence of these symptoms over different geographic areas. To achieve this, we make use of various big data tools such as Hadoop; Flume; Hive and Natural Language Processing tools such as TweetNLP. The primary reason behind using big data tools is that they can handle huge amounts of unstructured data streaming from multiple sources.

Our primary steps in this case study were the following:

- Set up a big data analytic and data management platform.
- Collect health-related social media data.
- Reduce noise using twitter-specific natural language processing and parts-of-speech (POS) tagging.
- Use clustering methods to group disease symptoms into relevant categories
- Visualize results of the analysis

These steps are described below in detail.

**Setting Up a Big Data Platform**

For this case study we chose to install the Apache Hadoop Big Data platform (The Apache Software Foundation, 2014). The Apache Hadoop platform is an open source product and one of the leading Big Data platforms used in both academia and industry. The Hadoop cluster we designed and configured consisted of a single master node and 6 data nodes. Next we streamed data from Twitter API to HDFS using Apache Flume. We also utilized TweetNLP (Owoputi et al. 2012) and Google Fusion Tables (Gonzalez et al. 2010); the former for noise reduction and the latter for visualization of geo-location enabled data.
Apache Hadoop

"Apache Hadoop is an open-source software for reliable, scalable and distributed computing" (The Apache Software Foundation, 2014). Hadoop can handle huge volumes of various types of data. This data can either be static or continuously streaming data. Hadoop distributes huge amounts of data and computational processes to a large number of low cost commodity machines (The Apache Software Foundation, 2014). It consists of two key components namely, HDFS (figure 1) and MapReduce. MapReduce provides Hadoop with capabilities such as mass storage and parallel computational power (Dean and Ghemawat 2008). HDFS is a distributed file system, which is designed to function on commodity hardware (Borthakur 2008). HDFS is also a highly fault-tolerant file system as compared with traditional distributed file systems. It provides high throughput access to applications that accesses very large datasets (Borthakur 2008). The platform also includes other sub-projects such as Apache Flume and Apache Hive (The Apache Software Foundation, 2014).

A typical hadoop cluster comprises of a single name node and multiple data nodes. All data on a Hadoop cluster are stored in the data nodes. Data stored in a Hadoop cluster are split into blocks of pre-defined size (default block size = 64 MB) and are stored in a specific number of data nodes based on a replication factor (default replication factor =3). Storing the same data block on multiple data nodes enhances data availability and reliability as the data block will be accessible even if a node containing it fails (The Apache Software Foundation, 2014).The name node keeps track of which data blocks of which files are stored in which data nodes. In this case study we use HDFS as the underlying storage for all our social media data and MapReduce jobs to access and analyze social media data in HDFS.

Apache Flume

Apache flume is an open source software used for collecting, grouping and moving large quantities of data (The Apache Software Foundatio 2014). Apache flume can be used to transport data such as social media data, log data, network traffic data and any other data sources. In this case study, we use Apache Flume to stream social media data from Twitter to the HDFS. Important components of a flume agent are source, channel and sink as shown in figure 2. A Flume source consumes events handed to it by an external entity such as a web server. A channel stores events until a sink consumes them. A sink puts the events from the channel into an external storage.
Apache Hive

Apache Hive provides an SQL-like interface to a large dataset that is stored in the HDFS (The Apache Software Foundation, 2014). Hive also allows the user to impose a structure onto the data and query the data using SQL-like language called HiveQL (The Apache Software Foundation, 2014). In this case study we used Apache Hive to structure data from twitter and prepare it for further data analysis.

Noise and TweetNLP

Tweets collected from Twitter are not always useful for analysis. Some keywords used in data collection can be used in many other contexts beside conversation on a disease condition. Tweets with such keywords are referred to as noise and render some collected tweets useless for analysis. We used a Natural Language Processing Tool called TweetNLP (Owoputi et al. 2012) to identify the context of the occurring keyword and to decide whether the context of occurrence of the keyword will be useful for our analysis. TweetNLP was developed by researchers at Carnegie Mellon University. It is a java based tokenizer and a parts-of-speech tagger for Twitter data. We use TweetNLP in this case study to tag social media data obtained from twitter in order to eliminate noise.

Google Fusion Table

Google Fusion tables aids in collecting, visualizing and sharing datasets over the web (Gonzalez et al. 2010). Data visualization in a Google Fusion table involves creating charts, maps and custom layouts in a short period of time. In this case study we use Google Fusion table to visualize geo-location enabled tweets. This is intuitive as it provides information about patterns of disease in a given geographic region in a given time interval.

Data Collection

To collect data from Twitter, we created a Twitter application. Once a twitter application was created, we received multiple authentication codes including a Consumer Secret, Consumer Token, Access Token and Access Token Secret from Twitter and used them for authentication. We also created a custom flume agent that would authenticate and collect data from the Twitter API based on certain keywords. This data was obtained in JSON format and was stored in HDFS. All these authentication parameters, keywords and storage path to be stored on HDFS were stored in a flume configuration file. For this case study we
identified some keywords that are related to certain disease conditions (diabetes, heart disorders, stroke, STD’s, flu, cancer) as well as treatments for these disease conditions. These keys words were generated based on a repository of health-related data hosted by two main sources. The first repository is Medline Plus, a service provided by the U.S. National Library of Medicine through the National Institute of Health (Medline Plus 2014). The second, MedicineNet, is a popular database that hosts medical information on various disease conditions (MedicineNet 2014). We collected health-related tweets for 10 days. This amounted to about 10GB of tweets. A section of the keywords we used in this case study are listed below:

Agina, Shortness of breath, irregular heartbeats, high blood pressure, stents, angioplasty, bypass surgery, pace maker, ace inhibitor, aspirin, beta-blockers, cough, rash, high potassium, numbness of face, numbness of arms, numbness of legs, loss of vision, loss of coordination, loss of speech, dizziness, ECG, EKG, clot dissolving medicine, aspirin, anti-platelet, loss of abilities, loss of motor skills, loss of speech, increased urination, fatigue, high blood glucose levels, insulin, loss of vision, kidney damage, nerve damage, fever, aches and pains, cough, runny nose, decongestant, immune stimulation, hyperthermia, vaccine, antitoxin, thymus extract, sonophoto dynamic, detoxification, cleansing, enema, chemotherapy, radiation, immune suppression, typhlitis, gastrointestinal distress, nausea, vomiting, anorexia, diarrhea, cramps

Data Preparation and Management

All the tweets collected from Twitter were in JSON format. A snapshot of the collected data is shown in Figure 3. This data included information such as username, timestamp for tweet, location data, re-tweet count, tweet language and time zone.

```sql
CREATE EXTERNAL TABLE Tweets (id BIGINT, created_at STRING,
```

---

**Figure 3. A Snapshot of Collected Twitter Data**

Since the JSON data was semi-structured, we created a schema for it using Apache Hive. Hive allowed us to structure our data like it would be in a traditional database and query the data using HiveQL. We initially started by creating a table for our tweets with columns including: id, created at timestamp, language of the tweet, re-tweet count, text and user. The HiveQL script for creating the table is shown below:

```sql
CREATE EXTERNAL TABLE Tweets (id BIGINT,
```
Once the data was structured we exported the structured data back to HDFS for analysis.

**Clustering Symptoms to Relevant Categories**

In order to perform further analysis on the tweets that had been collected and structured, we grouped keywords used to retrieve twitter data into categories that corresponds to specific disease conditions. This resulted in the creation of 10 groups representing disease conditions such as heart attack, stroke, kidney stones, cancer and asthma. There were some symptoms that were present in multiple groups. Symptoms and their corresponding groups are shown in Table 1.

After grouping the symptoms, we developed a clustering algorithm that read every tweet collected for the symptoms listed above and place the same in a corresponding group. This algorithm eliminates any tweet without keyword from one of the groups. The clustering algorithm was written in JAVA, based on Hadoop libraries. The methods in the clustering algorithm are listed in Figure 4. The following sub-section presents a description of the algorithm used.

- The `Main()` method is the main entry point for the clustering algorithm and gets executed first whenever the clustering algorithm is called. This method calls the following functions, which are significant in the data preparation process:
  - `clusterTweets()`: This method is used to cluster tweets based on the occurrences of keywords. This method also calls findCluster() and cleanText() methods in TweetClusterer class.
  - `folderVerifier()`: This method verifies whether the input path specified in HDFS is valid or not.
  - `getAllFilePathsInAFolder()`: This method is used to get all files stored under a folder in HDFS.
  - `getKeywordMatched()`: This method is used to return any keyword that matches with the tweet text.
  - `writeToCSVFile()`: This method writes clustered tweets onto local storage in CSV format.
<table>
<thead>
<tr>
<th>Group Number</th>
<th>Disease Condition</th>
<th>Corresponding Symptoms</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Heart Disease</td>
<td>Agina, Shortness breath, irregular heartbeats, high blood pressure, stents, angioplasty, bypass surgery, pace maker, ace inhibitor, beta-blockers, rash, high potassium, ECG, cardiac arrhythmia</td>
</tr>
<tr>
<td>2</td>
<td>Stroke</td>
<td>Numbness of face, numbness of arms, numbness of legs, loss of vision, loss of coordination, loss of speech, dizziness, EKG, ECG, clot dissolving medicine, aspirin, anti-platelet, loss of abilities, loss of motor skills, loss of speech</td>
</tr>
<tr>
<td>3</td>
<td>Diabetes</td>
<td>Increased urination, fatigue, blood glucose levels, insulin, loss of vision, kidney damage, nerve damage</td>
</tr>
<tr>
<td>4</td>
<td>Flu</td>
<td>Aches pains, cold, cough, fever, runny nose, decongestant, cough syrup, cough tablet, cough tablets</td>
</tr>
<tr>
<td>5</td>
<td>STD</td>
<td>Shingles, herpes</td>
</tr>
<tr>
<td>6</td>
<td>Diarrhea</td>
<td>Diarrhea, constipation, detoxification stomach, detoxification gut, detoxification body, cleansing stomach, cleansing gut, enema, gastrointestinal distress, typhlitis, nausea, vomiting</td>
</tr>
<tr>
<td>7</td>
<td>Tiredness</td>
<td>Anemia, fatigue</td>
</tr>
<tr>
<td>8</td>
<td>Muscle Spasm</td>
<td>Cramps</td>
</tr>
<tr>
<td>9</td>
<td>Liver Disease</td>
<td>Hepatitis, gall bladder, gall bladder stones</td>
</tr>
<tr>
<td>10</td>
<td>Cancer</td>
<td>Immune stimulation, hyperthermia, antitoxin, thymus extract, sono-photon dynamic, bio-electromagnetics, chemotherapy, radiation, immunosuppression, typhlitis, anorexia, malnutrition, neutropenia, sepsis, male pattern baldness, neoplasia, infertility, ovarian failure, neuropathy, cardiac arrhythmia, lymphoma, teratoma, leukemia, cardiotoxicity, hepatotoxicity, radiotherapy, fibrosis, tumor, bleeding cancer, lymph nodes, leukemia, carcinoma, breast cancer, colon cancer, prostate cancer, cervical cancer, melanoma, sarcoma, blastoma</td>
</tr>
</tbody>
</table>

Table 1. Symptoms and Their Corresponding Disease Groups

Figure 4. Clustering Algorithm Implementation I
Methods of the class Tweet Clusterer are used in data preparation and are shown in Figure 5.

- **findCluster()**: This method is used to find the appropriate cluster to which a tweet belongs.
- **cleanText()**: This method calls isNoise() in Noise Eliminator class. It is used to determine whether the tweet is relevant to our analysis or not.

![Figure 5. Clustering Algorithm Implementation II](image)

Figure 6 also describes the pseudo-code of the implemented clustering algorithm. Both noise elimination and clustering occur simultaneously. A detailed description of Noise Eliminator is in the following section.

```
(clusterTweets (tweet){

  1. **Clean the tweet.**
     cleaned_tweet = cleantext(tweet);

  2. **Find the cluster which the tweet belongs.**
     Cluster = findCluster(cleaned_tweet);

  3. **If the tweet doesn’t belong to any cluster terminate the function**
     If (cluster==null)
       Return;
     End if

  4. **Find whether the tweet is noise. If the tweet is noise return nothing else return the cluster which the tweet belongs.**
     is_tweet_noise = eliminateNoise(cleaned_tweet,cluster);
     If (is_tweet_noise) then
       Return;
     Else
       Return cluster;
     End if

}
```

Figure 6. Pseudo-code of Clustering Algorithm
Noise Reduction Using Natural Language Processing and Parts-of-Speech Tagging

There was considerable amount of noise in the collected tweets. This usually occurs when some keywords in the collected tweets are out of context and irrelevant to the conversation of interest. To aid in eliminating noise by removing irrelevant tweets, we developed a JAVA application that makes use of TweetNLP to eliminate noise and used it along with the clustering algorithm implementation described in the previous section. It identified the context of occurrence of keywords and aided in determining whether a tweet was useful for the analyses or not. The noise eliminator helped remove tweets that were not relevant to our subject of discussion. Methods of the noise eliminator we developed are shown in Figure 7.

- analyzePOSTaggerResults(): This method calls TweetNLP and it returns a tweet with its corresponding parts of speech (POS) tags.
- isNoise(): This method calls analyzePOSTaggerResults() method and distinguishes noise from tweets useful for our analysis based on the POS tags obtained from TweetNLP.

Analysis and Visualization of Results

In this section, we demonstrate the analysis and generation of results for a single disease group (Group 1). Group 1 consisted of tweets that had keywords related to heart diseases. These results can be extended and reproduced for the rest of the disease groups as well.

Figure 8, provides a summary of the number of tweets we collected, cleaned and analyzed. Initially we collected approximately 70 million tweets. After noise elimination, the number of tweets left for analysis was reduced to approximately 21 million tweets. Only about 1 million had geo-location information.

The pattern in which the keywords (belonging to Group 1 - a subset of cleaned dataset) are distributed for 10 days is shown in Figure 9. It is shown that keywords such rash, high potassium and high blood pressure are the most occurring terms related to heart disease. Researchers at Mayo Clinic have indicated that skin rashes in unusual spots is a heart disease symptom for heart infection (Mayo Clinic 2014). In fact, as is shown in figure 9, rash is the most prevalent disease symptom as observed in our data set. It is therefore a possibility that the users who posted these symptoms may be suffering from any of the three types of heart infections, namely; Pericarditis, Myocarditis or Endocarditis. The graph in figure 9 also
shows that one of the most common diagnostics tools used for heart diseases is the electrocardiography (ECG). This is consistent with treatment approaches in practice (Ferrari et al. 1997).

Figure 8. Number of Tweets Collected and Cleaned

Patterns in which keywords are distributed are shown on an hourly basis in Figure 10. Monitoring these keywords on an hourly or minute basis will provide in-depth insight into how symptoms progress and how diseases propagate. As a result, we can identify what the prominent symptoms leading to the disease are. For instance, we can determine if a symptom such as rash is reported consistently throughout the day or it is common on during certain times of the day. Figure 10 also shows that more symptoms are reported in the early morning and late night periods as compared to periods after midnight and mid-day.
Can Social Media Support Public Health?

Figure 9: Distribution of Keywords for Group (Heart Diseases) for Entire Period of Time

Figure 10: Distribution of Keywords for Group 1 (Heart Diseases) for an Entire Day (27-November-2013).
Can Social Media Support Public Health?

In Figure 11, the relationship between high blood pressure and ECG can be shown. A regression line is drawn to provide intuitive information on the interaction between these variables. The reason for the dip in ECG when there is an increase in high blood pressure can be attributed to the relationship of these keywords with other keywords. Similarly, relationships between all the other keywords can be studied.

In this case study, we were able to extract the geo-locations from the tweets that had them. We plotted the geo-locations that we extracted on Google Fusion Table (Gonzalez et al. 2010) and this can be seen in Figure 12. Geographical location is very vital in monitoring diseases as it sheds light on how the disease progresses in a given location at a given time. Geo-tagged tweets were plotted for a specific period of time. Geo-location information is vital to disease surveillance as it provides key information such as source of the outbreak, propagation speed and number of people affected per a particular area.

Figure 11: Analysis of Linear Relationship between High Blood Pressure and ECG

Figure 12: Plot of Geo-Location Enabled Tweets that Belong to Group 1 obtained between 00:00 Hours to 03:00 Hours on 28-November-2014) to a Geographical Map
Conclusion

In this case study we have demonstrated how data from social media could be very helpful in disease surveillance. With the help of big data technology, a huge amount of streaming data from the Twitter social media platform was analyzed. Also in this case study, we have shown how analyzing the distribution of symptoms of a particular disease can monitor disease propagation. This, coupled with geo-location data could provide valuable insights such as point of origin and propagation rate of disease symptoms. When this information is made available to relevant authorities, major disease outbreaks can be controlled. With intervention initiated at right time, lots of money and lives could be saved. Preliminary results presented have shown the potential of social media platforms a geo-located based source of data for disease surveillance. However, a limitation of this case study is the relatively small number of geo-tagged tweets as compared to the total tweet data set. This is because even though a large amount of tweets were collected, not all users had provided accurate location data as part of their profiles. In future studies, we would include secondary data about geographical location in our analysis so as to enrich our geo-spatial analysis. In terms of the domain of interest, we also intend to focus more on disease conditions that have the potential of breaking into epidemics and outbreaks as we believe such disease conditions would benefit more, in term of effective disease management, from close and constant surveillance and monitoring.

REFERENCES

papers3://publication/uuid/BE03DF70-D0C1-441E-A65F-1888C84992D6).


