Real Time Drug Safety Monitoring Project

**1. Vision and Goals of the Projects**

The Real Time Drug Safety Monitoring Project will be a real-time medical tool that will allow users to closely monitor and detect adverse side effects and patterns related to drugs. The project aims to crowdsource and mine large publicly available data sources in order to develop further knowledge on what people are saying about drugs and gain insight into the signs and symptoms that they encounter as well as the procedures performed. Several years ago, there was a Vioxx scandal in which more than 38,000 deaths were associated to usage of the drug, leading up to the drug being recalled. The goal of the Real Time Drug Safety Monitoring Project is to prevent cases like these from happening again.

Users/Personas Of The Project:

The Real Time Drug Safety Monitoring Project will be used by researchers and analysts who seek to document the effects of drugs in real-time, determine if there are negative side-effects associated with the drugs, and prevent life-endangering drugs from being available on the market.

It targets:

* Members of the medical community who are doing studies on the effects of drugs on patients.
* Researchers who want get more information can be learned from twitter, potentially supporting new health informatics hypotheses.
* Hospitals and research facility who wants to track the trends in medical conditions over time, analyze geographical factors linked health statistics from twitter, and combine these to track an ailment over time and geography.
* Discover ailments include both mider, acute illness as well as more common chronic conditions on people who do not visit their doctors, instead managing illness on their own.
* Research faculty and hospital who wants to track health status on people of all ages because of twitter user demographics. Most twitter users tend to be younger and only 2% are 65 or older.
* Analysis on the scale of individual users. It is less likely for a single user post multiple tweets, and multiple updates over the course of an illness.
* Collects finer grained location information. It is hard to obtain sufficient messages even on a state level because of the coarse geographic analysis.

**2. Solution Concept**

Global Architectural Structure Of the Project and a Walkthrough:

The system is expected to have three tiers which employ the following technologies:

* Apache Spark Streaming: a computing framework that will be used to gather tweets from Twitter
* Apache cTAKES/Analytics: an open-source processing system that uses machine learning methods(ATAM model) and allows users to extract information from electronic medical records. The end system will use cTAKES to retrieve clinically tested side effects and cross reference them with the tweets retrieved from Twitter.
* BigTop’s Spark Stack: an application which uses Spark Streaming in order to initiate a stream of twitter data and processes the data using cTAKES.
* OpenStack: a cloud operating system.
* NoSQL Database: an open-source relational database.

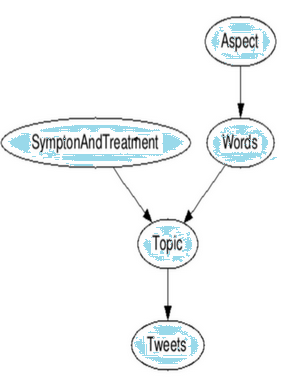


Figure1：Ailment Topic Aspect Model

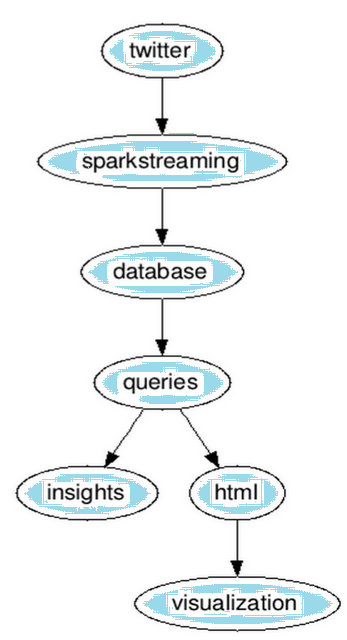


Figure 2: Picture Representation of Architectural Structure Of the Project

Design Implications and Discussion

Database to store both tweets and tweets’ analysis: While this will increase the amount of data that needs to be stored, it is necessary to store both the tweets and the tweet analysis in order to both have a record on the tweet analyzed, and have the analysis with it so a user could manually check if the tweet analysis was done correctly to a corresponding tweet. Using cTAKES to do this tweet analysis is a relatively untested field, so the early stages of this real-time monitoring will need to be repeatedly checked to ensure it is working as planned.

Use of cTAKES medical library: While there are other natural language processors out there, built into cTAKES is an extensive medical library that contains a wide variety of medical terms, including the names of various drugs and their synonyms. Instead of having to build our own database for drugs and necessary medical, cTAKES already has this so we will rely on this library and other functions cTAKES has to extract drug information from tweets.

Limited tweet scope for cTAKES to the 100 top-used drugs: Theoretically, we could scan Twitter and come across tweets for all kinds of drugs and the user-reported symptoms. By limiting our Twitter query to only the top used drugs, we improve the frequency at which tweets can be gathered relating to users posting information on drug symptoms. There will be many tweets gathered that will have nothing to do with drugs and symptoms of drugs, so this focused scope allows the program to collect more significant data faster.

**3. Scope and Features of the Project**

* Real Time Drug Monitoring System
  + Apache cTAKES
    - The primary driver of our project is cTAKES. cTAKES is a processing system for information extraction from electronic medical records.
      * Use Scala SBT.
      * Our application will access certain libraries from cTAKES such as various symptoms like nausea, coughing, et cetera.
      * Gather feeds of medical related tweets from output streams.
  + Apache Spark
    - Apache Spark exists as the wrapper to cTAKES.
      * As the framework behind making cTAKES run, we will implement appropriate Scala code in making Spark run with cTakes.
        + cTAKES works by spinning up two daemons: one thread to read streams from twitter and another thread to do processing.

We can control the stream reading by allocating frequency and amount of Tweets we get at each interval.

Our processing will be written by us in the future. It will contain algorithms in which Tweets are parsed for keywords containing negative effects resulting from medicine.

* + Openstack
    - We will use Openstack to scale our application so that quick and massive computation will work.
    - The reason why we are using Openstack in our application is scalability.
      * While a list of a few drugs may be easy to compute, as the number of drugs increases to large quantities, much more resources must be allocated to our application.
      * We will not be using Openstack in the preliminary design of our application, but we will need to implement Openstack functionality in the future sprints.

**4. Acceptance Criteria**

The minimum acceptance criteria is a graphical visualization of the data that works against a single OpenStack installation. The stretch goals are:

* Integrating keywords of Twitter Tweets to other relevant keywords.
* Record the occurrences of certain keywords.
* Extraction of significant trends from the keywords.
* Using OpenStack to run our compute engine in parallel.
* Visualize data graphically.

**5. Release Planning**

Detailed user stories and plans are on the Trello board:

<https://trello.com/b/Dmj0ssVX/real-time-drug-safety-monitoring-3-5>

Release #1 (due by Week 5):

User Stories: Environment Setup, Tweet Collection, Learning Scala

Environment Setup

(Spark) We will finalize environment setup for all members of our group. The cTakes Spark app should be able to be modified and contributed to by everybody. This means that everybody should be able to replicate the tutorial that Jay has created for us.

Tweet Collection

(cTakes) We will start collecting tweets using cTakes and writing code to figure out which tweets are useful. The app from Jay’s tutorial currently streams tweets. If we can output the output stream into a text file line by line, we will have a sample template in which to test our code against.

Learning Scala

(Scala) We will start learning how to code in Scala. While nobody is expected to be perfect in coding in Scala by the end of the sprint, everybody should be able to understand what the driver.scala in the app does for the most part.

Release #2 (due by Week 7):

User Stories: Grooming Problems from Last Week, Tweets to Cassandra

Grooming Problems from Last Week

(Spark and cTakes) We will groom user stories that occurred in the previous sprint. This may include setting up the environment for members of our group that are still struggling on getting the environment setup. This may include helping each other write Scala to modify the code.

Tweets to Cassandra

(Cassandra) We will write code that will take in each line of the tweets, find key words in the tweet, and write algorithms to deal with important tweets. The important keywords (such as nausea, cough, drug names, time of tweet), will be stored into Cassandra. This will be the most important aspect of our design as careful planning and coding will be necessary to make this a success.

Release #3 (due by Week 9):

User Stories: Visualization

Visualization

(To be determined) We will start researching and implementing visualization of our data. Ideally, we should be able to find some graphing utilities on Scala, but if those utilities do not exist, we will use graphing utilities on Java to do so.

Release #4 (due by Week 11):

User Stories: Relationships between Tweets, More Visualization

Relationships between Tweets

(To be determined) We should be connecting tables of keywords to other tables of keywords. For example, somebody who has been coughing for 3 days after taking a popular drug should be connected together by:

1. The Twitter user
2. The drug
3. The symptom, coughing

More Visualization

(To be determined) If perfect visualization of the graph is not completed yet, we will continue doing research for alternatives or improve the graphing framework we will have at this point.

Release #5 (due by Week 13):

User Stores: Optimization

Optimization

(To be determined) Ideally, the project should be close to being completed if not already completed. If it is already completed, we will focus on making our algorithm faster. We will also focus on making more sense of our data by displaying it in various different arrangements.