A Big Data System for Finding Reports of Drug-Drug Interactions on Twitter

Ian Wood

May 27, 2015

1 Introduction

Drug-drug interactions in patients taking many medications are a dangerous cause of morbidity [1]. However, due to the very large number of possible drug combinations, not every possible drug interaction can be tested in vitro, in vivo, or in a clinical trial. We are therefore interested in whether people are already self-reporting potential drug-drug interactions on social media. Using this information, we could help guide hypotheses on likely drug-drug interactions for further study. In particular, are users reporting both the drugs they are taking and symptoms they are experiencing on Twitter? Can we find any symptoms co-occurring with combinations of drugs more often than we should expect?

1.1 System Overview

In order to detect and analyze data from Twitter we will build the following system using components from the Apache Big Data Stack of open source software. The Futuresystems Openstack service is used to deploy virtual machines as the cluster of servers for our system. Chef, the devops software, is used to help with some installation. The Apache Hadoop framework with its distributed file system and MapReduce system is a dependency of the other pieces of the system. Zookeeper is used as a coordination service. Apache HBase is a distributed NoSQL database used for data storage. Apache Pig is a high-level language used for data analysis. Python is used for installation and final analysis. Java is used for the actual data collection program.

The system is organized on a cluster of three servers, although expansion to more servers should be straightforward. One server is used as a master for the rest when relevant, including the Hadoop NameNode, ResourceManager, and NodeManager, as well as the HBase HMaster. The master server also holds the Apache Pig installation for data analysis. The remaining servers run Hadoop DataNodes. All servers run an HBase HRegionServer and Zookeeper QuorumPeer.

Two dictionaries are used from Lin et al.'s 2010 BICEPP paper [2], using the 857 generic drug names as well as 107 adverse effects (symptoms) from the Australian Medicines Handbook. A streaming filter is created to collect all public tweets from Twitter's public streaming API that mention any of these terms. Using the free API, up to 1000 such terms can be tracked. The user id, text, creation timestamp, and the symptoms and drugs mentioned in the tweet are saved to an HBase table. This data is later processed through Pig scripts and further ipython scripts.

2 Installing and Running the System

To install the drug-drug interaction symptom detector follow the steps below. The full source code is available here: https://github.com/cloud-class-projects/drug-drug-interaction

2.1 Initialize VMs

On india.futuresystems.org or other OpenStack server, clone the github repository:

```
$git clone https://github.com/cloud-class-projects/drug-drug-interaction.git
```

Create three virtual machines (VMs) using the futuresystems/ubuntu-14.04 image, and associate three floating IP addresses with those VMs. A script is provided to help with the creation of the servers, named `drug-drug-interaction/SetupResources/createServer.sh`. This script requires three available floating IP address. If using the script, replace all the VM names with your own username and id number, replace the key name with your ssh key's name, and replace the floating-IP addresses with your floating IP addresses. Run the script to create the VMs and associate the floating IP addresses.

In the `drug-drug-interaction/SetupResources` folder of the git repo, create a file called pword.txt, which contains the password to the ssh key used to create the VMs. This is read by the setup script to connect to the VMs. Also edit the file `configServers.py` to include the names, private ip addresses, and public ip addresses of your VMs. The data associated with a particular VM should appear in the same location in each list. The first VM is particularly important, as it will be the master server, containing the Hadoop Namenode, all managers, and analysis code. In the rest of this procedure the VMs will be referred to as hadoop1, hadoop2, and hadoop3 in the order they are entered into `configServers.py`

2.2 Install Software

Once your VMs are ready, make sure you are in the `drug-drug-interaction/SetupResources` folder. Start a python shell (tested for python 2.7.9). Issue the following commands in order:

```
>from deploy_servers import *
>#Connect to the VMs and establish a session
>establishConnections()
>#Setup the VMs to allow communication with each other
>connectHosts()
>#Install Chef to help with Hadoop setup
>installChef()
>#Move necessary files and install Hadoop
>moveFilesAndSetupHadoop()
>#Install Zookeeper
>setupZookeeper()
>#Install HBase
>setupHBase()
>#Install Pig on the first server
>setupPig()
```

Note that installation will take some time and many messages will be printed. I find it is helpful to have a second ssh terminal open to hadoop1 to check on progress. Some particular things to watch for: `installChef()` may return before the `chef-repo` is fully set up. Either wait or try calling `installChef()` again before calling `moveFilesAndSetupHadoop()`. Often if something goes wrong during a step it can be fixed by calling the function again (although this is not necessarily the cleanest way to fix the problem). If the python shell needs to be terminated, the process can pick up where it left off by first calling `establishConnections()` and then the next step of

the process. To see if each step is successful the following should be true. After `connectHosts()`, the root user should be able to ssh into `hadoop1`, `hadoop2`, or `hadoop3` from any of the machines. After `installChef()`, a `chef-repo` folder with full permissions should be located in `/home/ubuntu`. After `moveFilesAndSetupHadoop()` the services (run `$jps`) `Namenode`, `NodeManager`, and `ResourceManager` should be running on the hadoop1, and the service `DataNode` should be running on the hadoop2 and hadoop3. After `setupZookeeper()` the service `QuorumPeerMain` should be running on every VM. After `setupHBase()` the service `HMaster` should be running on `hadoop1` and `HRegionServer` on all VMs. The function `setupPig()` finishes quickly, but the actual download and installation on the VM takes longer; it is done when a folder called Pig exists and the grunt shell can be accessed by calling `$/home/ubuntu/pig/bin/pig` after setting environment variables with `$source /home/ubuntu/.bash_profile`.

In my experience errors are most likely occur during the installation of HBase. I think this is due to the movement of files between india (or OpenStack server) and the VMs. Running `setupHBase()` a second time often fixes this problem. If that still does not solve the problem, ssh into hadoop1 and try to manually start HBase as root user. Run `source /home/ubuntu/.bash_profile` to set proper environment variables. Check which services are running, both HRegionServer and HMaster should run on hadoop1. They can be started by calling

```
/home/ubuntu/hbase/bin/hbase-daemon.sh --config /home/ubuntu/hbase/conf/ start
    $service
```

where `$service` is either "master" or "regionserver". If you notice problems connecting to ZooKeeper in the logs, try killing the zookeeper process (run `jps` to see its process id). Since it is supervised, it should start up again; try starting the HBase daemons after Zookeeper has restarted.

2.3 Collect Data

On `hadoop1`, install git with `$apt-get install git` and clone the drug-drug-interaction github repository as done for india or the OpenStack server used to setup the VMs. In the `GetTweets/conf` folder is a blank configuration file named `twitterConnectBlank.yml` you will need to connect to Twitter. Rename the file `twitterConnect.yml` and enter your Twitter API credentials (you can apply for free here: https://apps.twitter.com/), or I can provide my twitter app credentials for the purpose of evaluation.

Source the `/home/ubuntu/.bash_profile` file. Run the hbase shell to create the table for the twitter data:

```
$/home/ubuntu/hbase/bin/hbase shell
>create 'tweets', 'user_id', 'drug', 'symptom', 'creation_ts', 'tweet_text'
>quit
```

To collect the tweets we supervise an executable jar located in GetTweets. A built jar is provided in the repo for convenience. If building from source, build the jar with dependencies using Maven and java 1.6 (netbeans project files are included) and move the jar to the `GetTweets` folder and name the jar `GetTweets-1.0-SNAPSHOT-jar-with-dependencies.jar`.

Create a new window in a tmux session. In the `drug-drug-interaction` folder run:

```
$supervise GetTweets
```

You should see messages indicating the IDs of the tweets being collected. Allow this to run for some time. You can detach the tmux session but don't close the SSH session. For reasons I haven't been able to determine, tweet collection stops about an hour after the ssh session is closed.

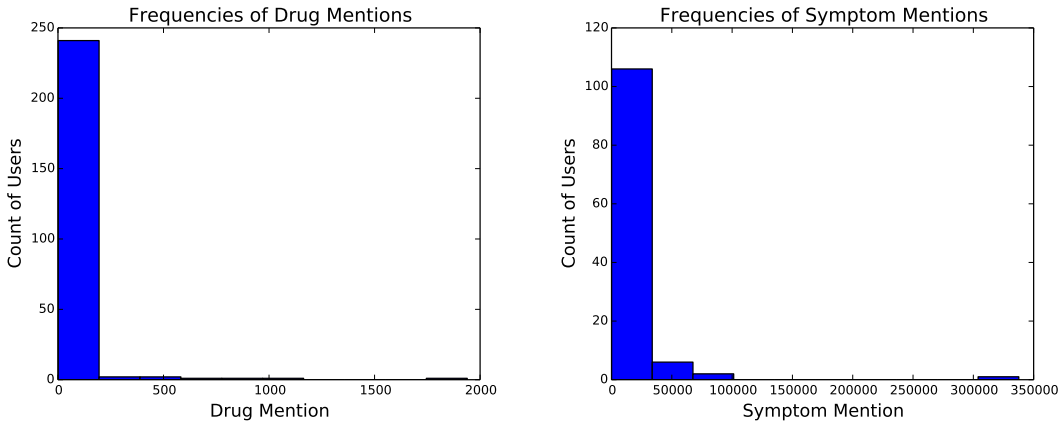

Figure 1: Frequency of Users Mentioning Drugs and Symptoms

2.4 Analyze Data

After tweets have been collecting for a while, run the analysis with Pig to count the number of users that have mentioned the drugs and symptoms in the dictionaries, also counting the number of users that have mentioned a drug and a symptom, two drugs, and two drugs and a symptom. As `root` user on `hadoop1`, run `$source /home/ubuntu/.bash_profile`. In `drug-drug-interaction/GetTweets` run the pig script wordCounts.pig with:

```
$/home/ubuntu/pig/bin/pig wordCounts.pig
```

This should exit without any failures. Once the pig script is finished, the results reside on the hdfs; collect them to a local folder and reorganize them to a more convenient format by calling `$./getResults.sh`. The `results` folder will now contain the counts and co-occurence counts of users mentioning drugs and symptoms as a series of csv files.

The ipython notebook `AssociationRules.ipynb` contains code for calculating measures from association rule learning for the user counts, in particular producing rules for the association between sets of co-occuring drug-symptom combos. Example results for over one million tweets and users are included in the results folder but will be overwritten by running `getResults.sh` and `AssociationRules.ipynb`.

3 Results

Over the course of three days I collected 1,467,848 tweets from 1,010,667 users that mentioned a drug or symptom in one of the dictionaries. As seen in Figure 1 most drugs and symptoms are rarely mentioned, although a few are mentioned very often. Similarly, as seen in Figure 2 co-occurring (in the same User's tweets) drug and symptom mentions similarly are rare with a few exceptions.

The top ten most frequently mentioned drugs, symptoms, and combinations are shown in Table 1, Table 2, Table 3, Table 4, and Table 5. The drugs most frequently mentioned are not very surprising. They include popular over-the-counter pain relievers; more powerful pain relievers; as well as cocaine, a well-known illegal drug; and dopamine, an important neurotransmitter. The most mentioned drug, however, is iron, which may refer to objects made out of the metal rather

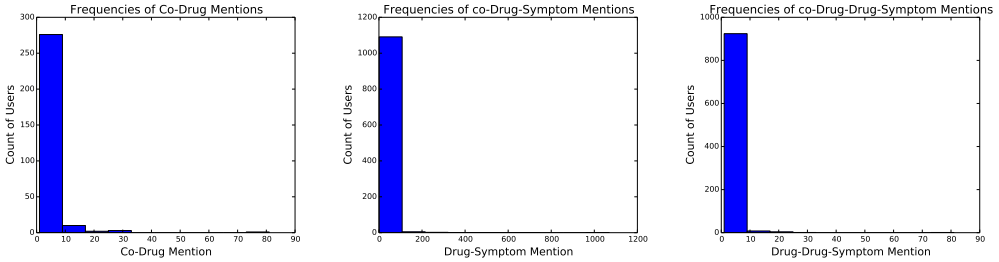

Figure 2: Frequency of Users (Left to Right): Mentioning two Drugs, mentioning a drug and a symptom, and mentioning two drugs and a Symptom

drug	count
iron	1939
nicotine	1157
paracetamol	803
caffeine	674
morphine	412
ibuprofen	412
aspirin	335
cocaine	217
acid	144
dopamine	101

Table 1: Top Mentioned Drugs

than the supplement. The high occurrence of acid is a mistake in the analysis. The tokenizer broke multi-word drugs into separate drugs during the analysis, like folic acid, salicyclic acid, and zoledronic acid, a problem which will need to be corrected in future versions.

It is more difficult to determine the relevance of the most mentioned symptoms to disease or drugs. The most mentioned symptom is "pain" which is included in many emotional or lyrical tweets, and doesn't necessarily refer to physical pain. Similarly weakness, burning, and mania, among others may not be medically relevant uses of the term. Consider the maxim, "pain is weakness leaving the body", commonly used to encourage exercise, which might occur alongside mention of "iron", commonly used in gym equipment and brand names.

The problem of tokenizing by white space is clearly seen in the top drug-drug combinations, shown in Table 3. However, we also see glucagon, milrinone, and dobutamine occurring together, as well as pain relievers co-occurring. Most of the drug-symptom combinations in Table 5 are directly between the drug and a symptom it treats, besides the many symptoms corresponding to iron that I believe are mostly about exercise.

The drug-drug-symptom combinations in Table 5 include more examples of multi-word drugs and the symptoms they treat, like zoledronic acid reducing the likelihood of fractures and salicyclic acid used to treat acne. The co-occurrences of dobutamine, milrinone, glucagon, and hypotension are due to retweets of a user named LearnTheHeart.com's tweet about reversing the symptoms of a beta-blocker overdose with the drugs.

In an attempt to find symptoms that appear more often with two drugs than would be expected if their occurrence with each drug individually was independent, we can turn to measures from association rule learning. In Table 6 the support, confidence, and lift (interest)

symptom	count
pain	337707
headache	81557
anxiety	78568
burning	65802
depression	65323
weakness	56353
nightmare	53246
mania	39464
fever	36378
stroke	28410

Table 2: Top Mentioned Symptoms

drug-drug	count
(acid,zoledronic)	81
(acid,salicylic)	30
(oil,peppermint)	25
(benzoyl,peroxide)	25
(ketamine,morphine)	19
(acid,folic)	18
(dobutamine,milrinone)	14
(ibuprofen,paracetamol)	14
(glucagon,milrinone)	14
(morphine,nortriptyline)	14
(acid,mefenamic)	14
(dobutamine,glucagon)	14

Table 3: Top Mentioned Drug Combinations

drug-symptom	count
(nicotine,pain)	1069
(paracetamol,pain)	692
(iron,mania)	622
(caffeine,headache)	436
(morphine,pain)	401
(iron,pain)	289
(aspirin,headache)	249
(ibuprofen,pain)	211
(iron,burning)	171
(iron,weakness)	148

Table 4: Top Mentioned-Drug Symptom Combinations

drug-drug-symptom	count
(acid,zoledronic,fractures)	81
(acid,salicylic,acne)	27
(benzoyl,peroxide,acne)	24
(ketamine,morphine,pain)	19
(oil,peppermint,headache)	17
(acid,folic,stroke)	17
(glucagon,milrinone,hypotension)	14
(dobutamine,milrinone,hypotension)	14
(morphine,nortriptyline,pain)	14
(dobutamine,glucagon,hypotension)	14
(acid,mefenamic,pain)	14

Table 5: Top Mentioned Drug-Drug-Symptom Combinations

of associations between drug-symptom combinations are reported. For a rule, $(X \Rightarrow Y)$, between sets X and Y: *support(X)* is defined the proportion of users in the dataset that mention an element of X, the support in the table is $support(X \cup Y)$; *confidence*$(X \Rightarrow Y)$ = $support(X \cup Y)/support(X)$, the proportion of instances of X that include elements of Y; and *lift*$(X \Rightarrow Y) = support(X \cup Y)/(support(X) \times support(Y))$, or the ratio of observed occurrences to those expected if X and Y were independent [3].

The top 20 rules by lift are included in the table, although because many rules have the same lift, this ends up being an arbitrary subset of the rules with the same lift. If ordered by support, by definition, the top rules would be between the elements of the most frequent drug-drug-symptom combinations. By focusing on lift we try to see combinations of drugs and symptoms that appear together more often than we would expect from the combination of each drug and symptom individually. Unfortunately, with very low support, these are largely due to a single user name MedEd101, a PharmD trying to spread knowledge about drugs and symptoms, so many drugs and symptoms are mentioned by the same user who takes none of them. The *(influenza, rash)⇒(sulfate, rash)* rule is an instance of a split drug name, namely the influenza vaccine and the rash it can cause.

4 Discussion

We find that many users do report the drugs they are taking and the symptoms they are experiencing on Twitter. However, we have so far not found convincing evidence of symptoms occurring more often with drug combinations than we would expect. While this project is a start to a large scale analysis of drug and symptom mentions on Twitter, there are a number of issues that must be addressed. The installation code is very complicated and should be improved to allow others to deploy these same tools, or even allow an individual researcher to install updates to these systems. The analysis must be improved in order to find relevant symptoms caused by drug-drug interactions. I detail these issues in the subsections *System Issues* and *Analysis Issues* respectively.

4.1 System Issues

The installation of the system could be greatly improved. In particular, the installation relies on hard-coded waiting times that are sometimes inadequate. A more sophisticated waiting method

Association Rule	Support	Confidence	Lift
(influenza,rash)⇒(sulfate,rash)	9.894e-07	1.0	1.011e+06
(alprazolam,seizure)⇒(amiodarone,seizure)	9.894e-07	1.0	1.011e+06
(alprazolam,stroke)⇒(metoprolol,stroke)	9.894e-07	1.0	1.011e+06
(alprazolam,seizure)⇒(metoprolol,seizure)	9.894e-07	1.0	1.011e+06
(alprazolam,hypertension)⇒(metoprolol,hypertension)	9.894e-07	1.0	1.011e+06
(alprazolam,stroke)⇒(hydralazine,stroke)	9.894e-07	1.0	1.011e+06
(alprazolam,seizure)⇒(hydralazine,seizure)	9.894e-07	1.0	1.011e+06
(alprazolam,tachycardia)⇒(hydralazine,tachycardia)	9.894e-07	1.0	1.011e+06
(alprazolam,hypertension)⇒(hydralazine,hypertension)	9.894e-07	1.0	1.011e+06
(alprazolam,stroke)⇒(dipyridamole,stroke)	9.894e-07	1.0	1.011e+06
(alprazolam,seizure)⇒(dipyridamole,seizure)	9.894e-07	1.0	1.011e+06
(alprazolam,tachycardia)⇒(dipyridamole,tachycardia)	9.894e-07	1.0	1.011e+06
(alprazolam,hypertension)⇒(dipyridamole,hypertension)	9.894e-07	1.0	1.011e+06
(alprazolam,stroke)⇒(theophylline,stroke)	9.894e-07	1.0	1.011e+06
(alprazolam,seizure)⇒(theophylline,seizure)	9.894e-07	1.0	1.011e+06
(alprazolam,hypertension)⇒(theophylline,hypertension)	9.894e-07	1.0	1.011e+06
(amantadine,confusion)⇒(bromocriptine,confusion)	9.894e-07	1.0	1.011e+06
(amiodarone,sedation)⇒(aspirin,sedation)	9.894e-07	1.0	1.011e+06
(amiodarone,dizziness)⇒(aspirin,dizziness)	9.894e-07	1.0	1.011e+06
(amiodarone,hypertension)⇒(aspirin,hypertension)	9.894e-07	1.0	1.011e+06

Table 6: Association Rules, including support for drug-drug-symptom combination, confidence of the rule, and the lift of the rule

is also used, wherein the installation script waits for printed responses after long downloads on the VMs, but this could also potentially fail, since no corrections are included for cases in which the "done" message is split between receiving buffers. In practice, I have never encountered this happening, but the potential for the program to enter an infinite loop exists.

The installation process is also highly dependent upon versioning, with versions coded into download links. The current process does not use the latest version of HBase, 1.0.0, but instead version (98.11). This is because the new version requires jdk1.7 while the hadoop and pig installations expect jdk1.6. In addition some of the ports seems to have changed in the newest version, such that it is more difficult to run a regionserver and master on the same VM. HBase 1.0.0 can be used, but requires manual tuning and is not supported by the installation script.

Due to changes in the python community as part of pip, urllib2, and other packages I was not able to utilize cloudmesh at the time I started my implementation. I confirmed with the team that the issues are not related to cloudmesh, but are in fact a temporary issue with the python community libraries that were used in cloudmesh. After consultation with the cloudmesh team I got confirmation from them that these issues have since be removed and the utilization of cloudmesh would have simplified the deployment issues while leveraging (a) the cloudmesh virtual cluster (b) the deployment of hadoop and other tools through the cloudmesh launcher. Functions for VM creation and automatic IP address collection and floating-IP address association using cloudmesh are included in `cloudmeshFunctions.py`, and can be used if cloudmesh is installed. The installation relies on open ports for a local network, using only local IP addresses. To use servers on different networks over public IP addresses security groups must be created with all relevant ports open for Hadoop, Hbase, Zookeeper, and Pig. I have not successfully installed the whole system with public IP addresses due, I believe, to port issues.

In addition, the data is not split over the regionservers. As the data grows, it should be split

or replicated periodically so that all data must not be transferred to each server during the Pig analysis.

4.2 Analysis Issues

The analysis is limited in a number of ways. First, white spaces must be properly replaced with underscores or other appropriate escaping method in order not to split apart multi-word drug names during tokenizing. Retweets must also be taken into account, perhaps ignored, to avoid artificially inflating the support for drug and symptom combinations. The current method of combining all drug and symptom mentions by a user over the entire dataset should also be limited, e.g. to a rolling time window, to focus on mentions that are more likely to be related to each other.

A more difficult problem will be to determine whether the symptom referred to is the purpose for taking the drug or a side-effect. In addition, educational users like LearnTheHeart.com, MedEd101 should be avoided in the analysis if possible. More techniques from natural language processing might be used to determine if a drug is taken for a symptom or if a user is actually taking a drug.

In addition, more data would lead to better analysis. The data reported here is a relatively small sample of the number of tweets every day. Larger dictionaries for drugs and symptoms would be useful, although tracking too many terms would run into the limit for Twitter's free API service.

References

[1] Kolchinsky, A., Lourenko, A., Li, L., and Rocha, L. M. (2013). Evaluation of linear classifiers on articles containing pharmacokinetic evidence of drug–drug interactions. In *Pac Symp Biocomput*, volume 2013, pages 409–420. World Scientific.

[2] Lin, F. P., Anthony, S., Polasek, T. M., Tsafnat, G., and Doogue, M. P. (2011). Bicepp: an example-based statistical text mining method for predicting the binary characteristics of drugs. *BMC bioinformatics*, 12(1):112.

[3] Tan, P.-N., Kumar, V., and Srivastava, J. (2004). Selecting the right objective measure for association analysis. *Information Systems*, 29(4):293–313.