Discovering New Drug-Drug Interactions by Text-Mining the Biomedical Literature

Bethany Percha

I. Introduction

Americans are living longer than ever before, and with that increased age comes a greater reliance on pharmaceuticals. For example, recent estimates by Kaiser Permanente indicate that the average 70-year-old American fills over 30 prescriptions per year [1]. The chance of an adverse drug reaction increases exponentially as each new drug is added to an individual's regime. What many people do not know is that clinical trials for new drugs do not typically address the issue of drug-drug interactions (DDI) directly, and often test new drugs in young, healthy people who are not part of a given drug's target population. Because of this, potentially-serious DDI are often not discovered until a drug is already on the market. In addition, a patient may be unaware that a symptom he experiences is due to a DDI, and may blame it on other factors. Many DDIs, therefore, probably go unreported.

Chemically-speaking, most DDIs are the result of one of two possible factors. First, a drug may inhibit an enzyme that is responsible for metabolizing another drug, effectively increasing the second drug's concentration in the body. And second, a drug may cause the body to produce more of an enzyme that metabolizes another drug, effectively decreasing that drug's concentration [2]. In both cases, the DDI is actually the result of both drugs' interacting with a single enzyme, which is the protein product of a gene. Therefore, most drug-drug interactions are actually drug-gene-drug interactions.

Unfortunately, while lists of known DDIs are widely available and commonly-used in clinical practice, drug-gene interactions are not as widely known. In addition, genes and drugs can interact in a variety of ways, and it is unclear which interaction types are most predictive of a drug's tendency to interact with other drugs. Furthermore, no complete databases exist that concisely describe the exact mechanisms by which drugs and genes interact; most of these interactions are only described in papers buried deep within the scientific literature.

In this environment, text mining presents a solution to the problem of uncovering novel DDIs. Previous work [7] has established methods for using a syntactical parser to identify and characterize drug-gene relationships. The end result was a semantic network of drug-gene relationships in which the edges consisted of several hundred interaction types normalized to concepts in an ontology. Here I present a method for using this approach to learn the types of drug-gene relationships that can predict drug-drug interactions, and then applying this method to predict novel DDIs.

II. Methods

A. Existing Code Base

Most of the code base that allowed me to extract semantic relationships for this project has already been constructed. However, the code has not yet been synthesized into a user-friendly pipeline, so the procedure took several steps:

- *Create two lexicons of terms, one for gene names and one for drug names.* I created two custom lexicons for this project. The first consisted of a set of 43 known pharmacologically-important genes identified by the PharmGKB database [3]. These were mostly liver cytochromes and various detoxification enzymes responsible for key processes within important metabolic pathways. The second lexicon consisted of 602 unique drug names obtained from a list of drug interactions provided by the Veterans Affairs hospital system, which meant that all were guaranteed to interact with at least one other drug in the lexicon.
- *Obtain a corpus of Medline article abstracts.* Fortunately, the Helix Group here at Stanford had already downloaded all Medline abstracts published before 2009. The corpus contained about 17.5 million abstracts and 88 million sentences.
- *Retrieve all sentences in Medline that mention both a drug and a gene of interest.* (For the purposes of this project, the drug and gene entities of interest will be known as *seeds*.) I accomplished this using my two lexicons and running 100 search processes in parallel on Stanford's BioX^2 cluster [4].
- *Represent sentences as parse trees using the Stanford Parser* [5]. If two seeds were not located in the same sentence clause, that sentence was removed from consideration. In addition, if a tree contained more than one clause and there was a clause that did not contain either seed, it was removed from consideration. A sample parse tree for one Medline sentence of interest is shown in Figure 1.

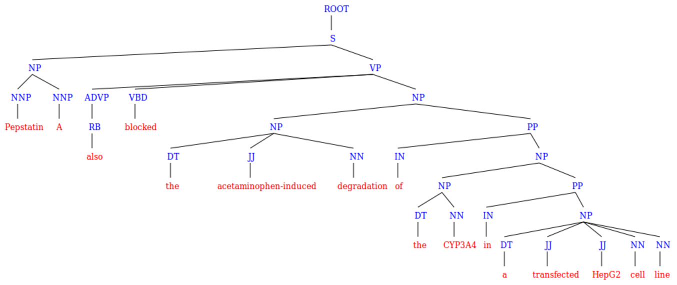

Fig. 1. **Parse tree for a single sentence in Medline.** The two seeds of interest are the drug name Pepstatin A and the gene name CYP3A4 (a liver cytochrome).

Department of Biomedical Informatics, Stanford University, Stanford, CA 94305–9510. E-mail: blpercha@stanford.edu. This paper was submitted as part of my final project for CS224N (Natural Language Processing) in the Winter 2011 term.

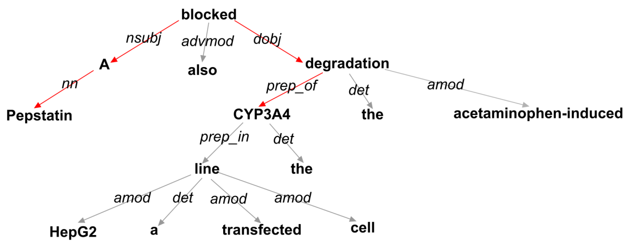

Fig. 2. **Dependency graph for the sentence shown in Figure 1.** The red arrows show the path through the graph that connects the seeds Pepstatin A and CYP3A4. Because this path contains the verb "blocked", this is a valid sentence. From here, it moves on to the normalization step.

- *Convert parse trees into dependency graphs, also using the Stanford Parser* [6]. The dependency graphs are rooted, oriented, and labeled graphs, where the nodes are words and the edges are dependency relations between words. The corresponding dependency graph for the parse tree in Figure 1 is shown in Figure 2.
- *Extract raw relationships between the two entities of interest.* Relations were of the form $R(a, b)$, where a and b represent the locations of the two seeds on the dependency graph, and R is a node that connects a and b and indicates the nature of their relationship. To make it past this stage of the analysis, the relation connecting seeds a and b must have been a verb (e.g. *associated*) or a nominalized verb (e.g. *association*).
- *Normalize relations.* This was the trickiest part of the analysis, and depended on a custom ontology developed by members of the Helix Group at Stanford [7]. The process of normalization entails mapping the raw relations onto a much smaller set of normalized relationships taken from the ontology. For example, the raw relations *associated* and *related* both map to the ontological entity *associated_with*. In addition, less-common terms like *augment* are mapped to more common synonyms, like *increase*. This has the advantage of decreasing the overall number of features that need to be considered in the analysis.

Based on my original lexicons of 602 drug names and 43 gene names, I was able to extract 9,418 sentences from Medline that contained two seeds of interest. All of these made it through the process of creating parse trees and dependency graphs, but only 3,522 made it through the process of extracting and normalizing the relations. Many sentences were cut because the relation connecting the two seeds of interest was not a verb or a nominalized verb, or because it was not a term recognized by the ontology. There were 344 unique relation types (*isAssociatedWith*, *induces*, etc.) represented among this final set of relations.

These 3,522 normalized relations represented the last part of the project for which I was able to use the Helix Group's previously-constructed code base. From here on, all of the analysis was conducted using my own [ugly] scripts in both R and Python.

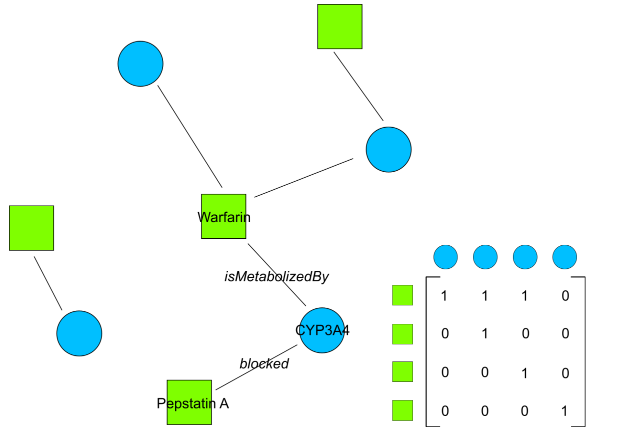

Fig. 3. **An example of a small semantic network.** The green squares represent drugs, and the blue circles represent genes. Note that the network is bipartite, meaning that there are no gene-gene or drug-drug connections; the only connections allowed are between a gene and a drug. The relation from Figure 2 is illustrated by the edge between Pepstatin A and CYP3A4, which is labeled with the relation name "blocked". Another hypothetical relation, "isMetabolizedBy", is shown between the drug Warfarin and CYP3A4. We might hypothesize from this graph that Pepstatin A and Warfarin would interact, since both are connected to the same gene. An alternative representation of this network is an adjacency matrix, an example of which is shown at right.

B. Building the Semantic Network

To learn the types of gene-drug interactions that were most predictive of DDI, I built a bipartite network like the one in Figure 3, in which the nodes were the gene and drug seeds from the two lexicons and the edges were labeled with the normalized relation types found in the literature. In Figure 3, the green squares represent drugs and the blue circles represent genes. The final network therefore consisted of 602 drug nodes and 43 gene nodes, with 3,522 gene-drug edges. The edges were labeled using the 344 different relations that appeared in the final dataset.

Computationally, I represented this network as a three-dimensional *adjacency matrix* of dimensions

$$\begin{array}{ccccc} 602 & \times & 43 & \times & 344 \\ \text{drugs} & \times & \text{genes} & \times & \text{relations.} \end{array}$$

The elements of the matrix, A_{ijk}, were 1 if there was ≥ 1 valid sentence in the literature connecting drug i and gene j via relation k, and 0 if no such sentence existed.

C. Learning Interaction Rules

My overall goal in this project was to learn the types of relations between genes and drugs that were most predictive of drug-drug interactions. Conceptually, this meant considering all two-edge paths through the network that connected two drugs via a gene

$$\text{drug}_1 \xleftarrow{\text{relation}_1} \text{gene} \xrightarrow{\text{relation}_2} \text{drug}_2$$

and determining which pairs of relation types were most indicative of drug-drug interactions. The number of paths of length 2 that include relation types m and n between drug i and drug j is given by

$$(A_m A_n')_{ij} + (A_m A_n')_{ji}$$

where A_m and A_n are two-dimensional 602×43 "slices" of the larger 3-dimensional adjacency matrix that correspond to relations m and n. By multiplying these matrices, we eliminate all information about exactly *which* gene(s) the paths pass through; we only care whether paths exist that encompass the relation types of interest. Likewise, we do not differentiate between

$$\text{drug}_1 \xleftarrow{\text{relation}_1} \text{gene} \xrightarrow{\text{relation}_2} \text{drug}_2$$

and

$$\text{drug}_1 \xleftarrow{\text{relation}_2} \text{gene} \xrightarrow{\text{relation}_1} \text{drug}_2.$$

The actual rule-learning process could be accomplished using a variety of supervised learning techniques in which the response variable was

$$y = \begin{cases} 1 & \text{drugs interact} \\ 0 & \text{no interaction,} \end{cases}$$

and the predictor variables, x_i, were

$$x_i = \begin{cases} 1 & \text{drugs connected by path type } i \\ 0 & \text{not connected by path type } i. \end{cases}$$

Since there were 344 unique relations and each path contained two relations, the total number of features considered in the analysis was

$$\begin{array}{ccccc} \frac{1}{2} \cdot 344 \cdot 343 & + & 344 & = & 59,340. \\ \text{edges different} & + & \text{edges the same} & & \end{array}$$

The total size of the training set was the total number of drug pairs, or $602 \cdot 601/2 = 180,901$. The number of known interacting drug pairs was $2,217$, so the data were quite sparse. The final dataframe used for the analysis consisted of 180,901 rows (all drug pairs) and 59,341 columns (59,340 features, plus one response column). I therefore represented it using the MatrixMarket sparse matrix representation format in R to increase computational efficiency.

Obviously, with so many features, some feature selection was required. I began my analysis by performing univariate t-tests for each feature, comparing its mean rate of occurrence between drug pairs that interacted and those that did not. Because my primary interest was in features that occurred more often for the interacting drug pairs than the non-interacting drug pairs, I used a one-sided t-test and only accepted features where the proportion of occurrences was greater in the interacting group. After performing a simple Bonferroni correction [8] for multiple hypothesis testing, the p-value cutoff for a given feature to be included in the final analysis was 8.432×10^{-7}. I then incorporated the features that survived this initial cut into a multivariate logistic regression model, which I used to predict whether other drug pairs would interact.

D. Predicting New Interactions

The power of this approach is that it allows us to use observed paths within the semantic network to predict previously-unknown drug-drug interactions. To evaluate the final logistic regression model's predictive power, I built the model using only 90% of the original data (chosen randomly) and then tested it on the remaining 10%. I evaluated the model's performance at choosing the drug pairs that were part of the original set of 2,217 interactions, as well as the proportions of false negatives and false positives that occurred in the test set analysis. Finally, I searched Drugs.com, a popular source of information about drug interactions, for information on the interactions predicted by the model that were not in the original list. I wanted to see if they were truly novel predictions, or if [more likely] they were known interactions that simply had not appeared in the list provided by the VA administration.

III. Results

A. Most Important Relations

As a first step in my analysis, I conducted t-tests for each feature - comparing its frequency of occurrence for interacting drug pairs vs. noninteracting drug pairs - and chose the features with the 100 lowest p-values. I then constructed a tag cloud of all the relations found within those features. The tag cloud is shown in Figure 4. The relations that occurred most often in paths linking interacting drug pairs were *isMetabolizedBy* and *isAssociatedWith*, closely followed by *induces*, *influences*, and *inhibits*.

The goal of this part of the analysis was simply to see if the feature extraction method was pulling out relations that looked reasonable. Since it is likely that two drugs metabolized by the same gene would interact, or that two drugs that induce production of the same protein would interact, these terms make intuitive sense.

B. Feature Selection

Unsurprisingly given the p-value adjustment for multiple hypothesis testing, there were only 9 features for which the proportion of occurrences among the interacting drug pairs was significantly greater than the proportion among noninteracting pairs. In addition, there were a few features that appeared/disappeared on this list depending on the

Relation 1	**Relation 2**	**P-value**
inhibits	isMetabolizedBy	1.35e-09
isAssociatedWith	isMetabolizedBy	3.67e-09
includes	isMetabolizedBy	3.25e-08
induces	isMetabolizedBy	4.32e-08
induces	isAssociatedWith	6.49e-08
inhibits	isAssociatedWith	2.30e-07
isAssociatedWith	isEvaluated	2.93e-07
has	isMetabolizedBy	3.91e-07
isAssociatedWith	isAssociatedWith	5.46e-07

Fig. 5. **Top 6 relations with P-values.** This list contains the six features most strongly associated with interacting drug pairs vs. noninteracting drug pairs.

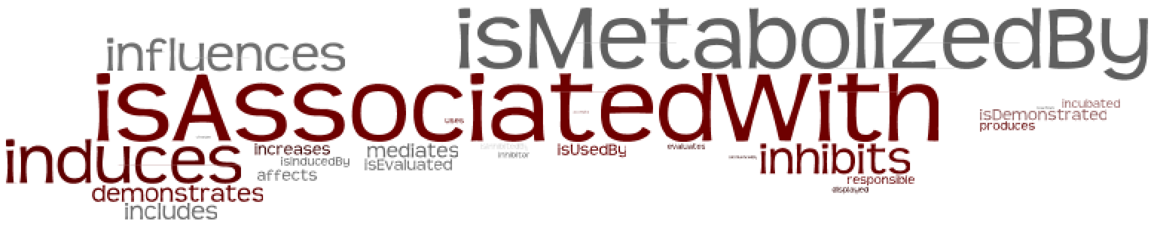

Fig. 4. **Tag cloud of most important relations.** The relations shown here are sized relative to how often they appeared in the top 100 features most predictive of drug-drug interactions.

specific 90% random sample chosen from the training set. The “consensus list” is shown in Figure 5.

The lowest p-value occurred for the feature

$$\text{drug}_1 \xrightarrow{\text{inhibits}} \text{gene} \xleftarrow{\text{isMetabolizedBy}} \text{drug}_2$$

which indicates that two drugs are likely to interact if one inhibits the production of a gene product that in turn is responsible for metabolizing the other. This makes perfect sense biologically, because co-administration of those two drugs would lead to highly-elevated levels of drug 2 within the body. Another important feature of interest is

$$\text{drug}_1 \xrightarrow{\text{induces}} \text{gene} \xleftarrow{\text{isMetabolizedBy}} \text{drug}_2$$

which would work in the opposite direction if drugs 1 and 2 were co-administered: the presence of drug 1 would induce the production of a gene product that metabolizes drug 2, leading to decreased levels of drug 2 within the body and a decrease in drug 2’s therapeutic efficacy. Some of the features, such as

$$\text{drug}_1 \xrightarrow{\text{isAssociatedWith}} \text{gene} \xleftarrow{\text{isMetabolizedBy}} \text{drug}_2$$

are less clear, mechanistically-speaking. The relation *isAssociatedWith* is a relatively high-level term in the ontology, and encompasses a wide variety of other terms that cannot be mapped to a lower-level, more-specific relation like *induces* or *inhibits*. Therefore, these interactions could very well represent biological mechanisms similar to the two discussed above; they simply weren’t described that way in the literature.

One interesting feature that placed highly on the list was

$$\text{drug}_1 \xleftarrow{\text{includes}} \text{gene} \xleftarrow{\text{isMetabolizedBy}} \text{drug}_2.$$

On the surface, it is unclear what the verb *includes* is referring to. Does the protein product of the gene include a molecule or structural motif that resembles the drug? It is difficult to tell without looking at the raw sentences. Upon further inspection, we see that *includes* is usually used in sentences directly describing relationships between drugs and genes, such as

```
Clinically important CYP3A4 inhibitors include
itraconazole, ketoconazole, clarithromycin,
erythromycin, nefazodone, ritonavir and grapefruit
juice.
```

This single sentence includes three drugs of interest and one gene of interest; given the wide variety of drugs metabolized by CYP3A4, therefore, it is no wonder that a feature comprised of *includes* and *isMetabolizedBy* is such a strong predictor of drug interactions. It is also worth noting that the real relations described in this sentence are

$$\begin{aligned} \text{erythromycin} &\xrightarrow{\text{inhibits}} \text{CYP3A4} \\ \text{nefazodone} &\xrightarrow{\text{inhibits}} \text{CYP3A4} \\ \text{ritonavir} &\xrightarrow{\text{inhibits}} \text{CYP3A4} \end{aligned}$$

so although the normalized relation chosen by the code was *includes*, the real relationships are inhibitory ones similar to the other features discussed earlier.

C. Predicting New Interactions

Using the nine features shown in Figure 5, I built a logistic regression model using known DDI status as the outcome and a random sample of 90% of the original data as the training set. I then used that model to predict which drug pairs in the test set were most likely to interact. Unfortunately, only three drug pairs in the test set were predicted to interact; that is, the probability of their interaction, as given by the logistic regression model, was greater than 0.5. These three pairs were:

```
ketoconazole tacrolimus
 erlotinib erythromycin
amlodipine erythromycin
```

The first was a known interaction from the list provided by the VA, but the other two were not on the list. The total number of known interactions in the test set was 217, so this seemed to indicate extremely poor model performance on this test set.

However, when I looked up the other two interactions on Drugs.com, I was surprised to find that one (erlotinib and erythromycin) was considered a moderately important interaction. For example, the following warning was issued for erlotinib:

> Caution is advised if erlotinib must be used with potent CYP450 3A4 inhibitors such as itraconazole, ketoconazole, voriconazole, nefazodone, delavirdine, protease inhibitors, and ketolide and certain macrolide antibiotics... According to product labeling, coadministration with the potent CYP450 3A4 inhibitor ketoconazole increased erlotinib area under the plasma concentration-time curve (AUC) by two-thirds compared to administration of erlotinib alone.

Indeed, erythromycin was one of the macrolide antibiotics known to interact with erlotinib. Erlotinib (brand name Tarceva) is a drug most often used in cancer chemotherapy, so its interaction with a drug used to treat bacterial infections is somewhat surprising. Nonetheless the model, based solely on the drugs' relationships to common genes as described in the scientific literature, was able to pick it up.

The other drug combination, amlodipine and erythromycin, was not listed as a known interaction on Drugs.com.

D. Further Predictions

I was interested in seeing which drug pairs my model ranked most highly as likely interactions, even if the probability of those interactions, as given by the model, did not reach the cutoff of 0.5. I therefore ranked the top 20 most likely interacting pairs from the test set, as predicted by the final model. The results are shown in Figure 6.

	Prob		
Drug Names	Interact	VA	Drugs.com
ketoconazole tacrolimus	0.719	X	X
amlodipine erythromycin	0.603		
erlotinib erythromycin	0.603		X
itraconazole ketoconazole	0.492		
gefitinib testosterone	0.225		
codeine quinidine	0.206		X
ketoconazole nefazodone	0.178		X
nefazodone tacrolimus	0.178	X	X
alprazolam nefazodone	0.178	X	X
nefazodone repaglinide	0.178		X
nefazodone pimozide	0.178	X	XX
gefitinib pravastatin	0.170		
amlodipine itraconazole	0.155	X	XX
fluoxetine terfenadine	0.116		XX
erythromycin methadone	0.115		XX
erythromycin itraconazole	0.115	X	XX
erythromycin midazolam	0.115	X	X
atomoxetine methadone	0.115		X
methadone nicotine	0.115		
captopril enalapril	0.115		

Fig. 6. **The top 20 most likely interacting drug pairs**, as predicted by the final model. Although many of these drug pairs were not represented on the original VA interaction list, they had at least moderate interactions according to Drugs.com. A single "X" represents a moderate interaction on Drugs.com, while "XX" represents a severe interaction.

Of the 20 drug pairs on the list, 7 (35%) had known interactions from the VA list and 14 (70%) had known moderate or severe interactions according to Drugs.com. However, these results only show that the model is able to pick up known interacting drug pairs at a fairly high rate. Since the drugs from the lexicon are known to interact with at least one other drug, the model may simply have relatively high sensitivity but low specificity (i.e. it is unable to tell when a drug pair will *not* interact).

To get a sense of the model's specificity, I chose a random sample of twenty drug pairs for which the probability of interaction (according to the model) was less than 0.02. I then repeated my analysis for those pairs. The results are shown in Figure 7. Of the 20 pairs, 2 (10%) were on the VA list and 7 (35%) had known interactions according to Drugs.com.

	Prob		
Drug Names	Interact	VA	Drugs.com
bexarotene potassium	0.012		
pancuronium penicillin	0.012		
indomethacin potassium	0.012		
miconazole tocainide	0.012		
hyoscyamine moxifloxacin	0.012		
bepridil fluoxetine	0.012		
cisplatin isoniazid	0.012		X
medroxyprogesterone rifapentine	0.012	X	X
lamotrigine triamterene	0.012		
insulin theophylline	0.012		
omeprazole valsartan	0.012		
mibefradil minocycline	0.012		
atorvastatin fluconazole	0.012	X	XX
felbamate prochlorperazine	0.012		X
hydrocodone memantine	0.012		
mephentermine metformin	0.012		X
heparin oxyphenbutazone	0.012		X
bumetanide sulindac	0.012		X
halazepam naproxen	0.012		
amobarbital isoniazid	0.012		

Fig. 7. **A random sample of drug pairs with interaction probabilities of less than 2%.** The symbols are the same as those in Figure 6.

IV. Discussion

In this report, I describe a novel method for predicting drug-drug interactions based on a combination of techniques from natural language processing and machine learning. The raw extraction of textual features of interest (the normalized relations) was accomplished using the same sentence parsing techniques we explored in Programming Assignment 3. I then used basic techniques from network theory (the concept of an adjacency matrix; using an adjacency matrix representation to find all paths of length 2 in a network) and machine learning (feature selection; the Bonferroni correction; logistic regression) to evaluate the textual features and find those most predictive of drug-drug interactions. To me, this project perfectly illustrates the power of natural language processing: distilling free text into machine-readable features that many known algorithms already know how to handle. This pipeline allows us to make meaningful inferences from text that would be difficult or impossible without first deconstructing the role of each textual element and deciphering how the different elements - noun phrases, verbs, etc. - relate to each other.

Of course, the performance of the final model on the test set was not ideal. If we consider the list of known interactions from the VA as our gold standard, the test set contained 217 interacting pairs and 17,874 noninteracting (or unknown) pairs. The model was only able to detect one of the 217 pairs; there were two false-positives and 216 false-negatives, along with 17,872 true negatives. If we estimate the model's sensitivity and specificity based solely on this test set, therefore, we obtain:

$$\text{sensitivity} = \frac{1}{1+216} = 0.0046$$

$$\text{specificity} = \frac{17872}{2+17872} = 0.9999,$$

which indicates that the model is great at rejecting drug pairs that do not interact, but terrible at picking up those that do.

There are several reasons for this that could be addressed in future versions of the model, however. First, the number of textual co-occurrences of terms from the two lexicons was actually quite small (9,418 sentences; 3,522 normalized relations) compared to the number of terms involved (602 drugs and 43 genes). The main reason for this was that my two lexicons did not include synonyms - I wanted to obtain preliminary results for the project as quickly as possible, and a full search of the literature that included all potential synonyms for this many gene and drug names can take over a week. To give a sense of how drastically this cut down my number of "hits", here is a list of all the synonyms for erythromycin, the macrolide antibiotic discussed earlier:

```
erythromycin;Dumotrycin;E-Base;E-Glades;E-Mycin;
E-Solve 2;EM;EMU;Eryderm;ETS;Erygel;Emgel;Erymax;
Eritrocina;Erypar;Erythra-Derm;Erythro;
Erythro-Statin;Erythromycin oxime;Erythrocin;
Ethril 250;Erythrocin Stearate;Ilocaps;Erythrogran;
Ilosone;Ilotycin;Ilotycin Gluceptate;IndermRetcin;
Abboticin;Stievamycin Forte Gel ;Kesso-Mycin;
Abomacetin;Stievamycin Gel ;Mephamycin;Ak-mycin;
T-Stat Lot ;Pantomicina;Akne-Mycin;T-Stat
Pad-Lot; Aknin;Taimoxin-F;Benzamycin;Theramycin
Z; Benzamycin Pak;Torlamicina;Bristamycin;Wemid;
Dotycin;Wyamycin S;Ermycin;Ery-Sol;Ery-Tab;Eryc;
Eryc 125;Eryc Sprinkles;Erycen;Erythroguent;
Erycette;Erythromast 36;Erycin;Erythromid;
Erycinum;Erythromycin A;Erythromycin B;
Erythromycin Stearate;Erythromycin estolate;
Pce;Erythromycin ethylsuccinate;Pfizer-e;
Erythromycin glucoheptonate;Propiocine;
Erythromycin lactobionate;R-P Mycin;Robimycin;
Sans-Acne Solution ;Sansac;Serp-AFD;Staticin Lot;
Stiemycin
```

Searching only for "erythromycin", therefore, misses many synonyms that would have been mapped to the term "erythromycin" during the normalization process. Perhaps more crucially, genes also have many synonyms. Here is a list for CYP3A4, a liver cytochrome discussed earlier:

```
CYP3A4;CYP3A;CYP3A3;CYP3A4*1;CYP3A4*1A;HLP;MGC126680;
NF-25;P450C3;P450PCN1;*1A;CP33;CP34;cytochrome
P450, family 3, subfamily A, polypeptide 4;
glucocorticoid-inducible P450;nifedipine oxidase;
P450-III, steroid inducible;cytochrome P450,
subfamily IIIA (niphedipine oxidase), polypeptide
3; cytochrome P450, subfamily IIIA (niphedipine
oxidase), polypeptide 4
```

Gene nomenclature is perhaps even less standardized than drug nomenclature, so eliminating synonyms for genes was an even more serious omission. In any case, this problem is easily rectified by including synonyms in subsequent literature searches.

On a related note, the model's lowest assigned probability of interaction (Figure 7) occurred for drug pairs where neither drug appeared anywhere in the literature. Since this total absence from literature references is unlikely, its most probable cause is that the search term was actually a less-common synonym for another entity. This could help explain why several interacting drug pairs showed up in Figure 7: the model simply did not have any evidence whatsoever about those pairs, and so coult not differentiate them from other pairs that truly did not interact.

In addition, the number of drug interactions from Drugs.com that did not show up in the VA's "gold standard" list is somewhat troubling. It seems clear that the VA's list is incomplete, and that a better gold standard will be necessary for future models. Unfortunately, obtaining a complete list of drug interactions is fairly difficult, since prominent web services like Drugs.com and Epocrates are usually careful to guard their raw data.

Finally, some of the model's poor performance might be the result of the original Medline corpus used when extracting the raw relations. This corpus included all abstracts from before the year 2009, but the biomedical literature is one of the fastest-growing bodies of free text in existence, and thousands of new abstracts are added every year. Much more information on drug-gene interactions is available now than was available two years ago. Subsequent work on this topic, therefore, may require a more current "edition" of the Medline corpus.

V. A Note on the Code

Most of the code for this project was provided by Yael Garten and Russ Altman of the Helix Lab, and was written by Adrien Coulet, a former graduate student who is now a professor in Switzerland. Going from raw Medline sentences to normalized relations was no easy task, and the code reflected this; it was an unholy combination of Bash scripting, random bits of Perl, and about a dozen Java classes, all of which had file locations hard-coded into them. For this reason (and because the lab is still using this code and I didn't have permission to share it) I am not including the code used for extracting the parse trees and normalized relations with my submission. I am, however, including the raw data files obtained from running their code on the BioX^2 cluster.

I did most of my own analysis in Python and R, and my scripts for that are included. They aren't pretty, but you can get a sense of what I did to create the adjacency matrices for the network and extract the relevant features. If you need any further information about the code, please don't hesitate to contact me.

References

[1] http://www.statehealthfacts.org/ Accessed Monday, March 7, 2011.

[2] Katzung BG, Masters SB, Trevor AJ *Basic and Clinical Pharmacology.* McGraw-Hill: New York, NY, 2009.

[3] T.E. Klein, J.T. Chang, M.K. Cho, K.L. Easton, R. Fergerson, M. Hewett, Z. Lin, Y. Liu, S. Liu, D.E. Oliver, D.L. Rubin, F. Shafa, J.M. Stuart and R.B. Altman, "Integrating Genotype and Phenotype Information: An Overview of the PharmGKB Project", The Pharmacogenomics Journal (2001) 1, 167-170.

[4] The Bio-X^2 cluster is the result of an NSF-funded research proposal submitted by 21 Bio-X affiliated faculty, representing 13 departments and 4 schools at Stanford. The purpose of the cluster is to facilitate biological research problems ranging in scale from molecules to organisms. It was funded by the National Science Foundation. The hardware represents generous donations by both Dell and Cisco.

[5] Dan Klein and Christopher D. Manning. 2003. Accurate Unlexicalized Parsing. Proceedings of the 41st Meeting of the Association for Computational Linguistics, pp. 423-430.

[6] Marie-Catherine de Marneffe, Bill MacCartney and Christopher D. Manning. 2006. Generating Typed Dependency Parses from Phrase Structure Parses. In LREC 2006.

[7] Adrien Coulet, Nigam H Shah, Yael Garten, Mark A Musen, Russ B Altman: Using text to build semantic networks for pharmacogenomics. Journal of Biomedical Informatics 43(6):1009-19 (2010)

[8] Miller, Rupert G. (1981) *Simultaneous statistical inference.* 2nd ed. Springer Verlag, pages 6-8.