Simon Zheng

BIOC 218

Spring 2012-2013

A Critical Review of Natural Language Processing Approaches to Discovering Drug-Drug Interactions from Medical Literature

Abstract

The objective of this paper is to provide a critical review of the natural language processing (NLP) approaches to discovering hidden drug-drug interactions (DDIs) from biomedical research literature. In this paper, I provide an overview of three techniques for using text mining to find DDIs in huge corpuses of research papers. The first technique to be presented involves using sentence co-occurrence in full-text articles [1], the next involves creating a template-based classifier for use in full-text articles [2] and the last uses text-mining and automated reasoning in research abstracts [3]. For each NLP approach, an overview of the technique is reviewed and followed by a discussion of the trade-offs between the different approaches and potential avenues of improvement.

There exist a number of other techniques but I chose to review these three because they represent some of the first and most popular attempts at using text-mining to discover DDIs. However, as the field of applying NLP to medical text is very new, other techniques generally either tweak and improve on these methods or bring in new methods from Machine Learning or NLP. This paper concludes with an overview of the three aforementioned techniques and notes the current progress in the field.

Introduction

Drug-drug interactions are defined as situations in which the simultaneous consumption of two or more drugs causes one or more of the drugs to affect the activity of another. Typically, drug-drug interactions occur when two drugs interact with the same gene product, though a

number of other modes of interaction exist, such as one drug changing the pH of the stomach, reducing absorption of another drug, among many other ways [2]. Historically, drug-drug interactions have been largely unpredictable and in 2011, it is estimated that DDIs account for nearly 30% of the 700,000 adverse drug event (ADE) injuries that occur every year [4].

However, with the steadily increasing number of drugs on the market and an aging population with growing medical needs, the study of drug-drug interactions has become more important in increasing longevity and quality of life. According to the National Health and Nutrition Examination Survey, over 76% of elderly Americans are on two or more drugs today [4] and the Kaiser Family Foundation indicates that the average 70-year-old American fills over 30 prescriptions per year [2]. In spite of the growing need for knowledge about DDIs, our understanding of them has been relatively murky, as clinical trials for new pharmaceuticals focus on establishing the safety and efficacy of *single* drugs and do not typically investigate DDIs [5]. For the past few decades, attempts to uncover these DDIs have been done by manual curation, which is expensive, slow and unscalable.

In the past few decades, however, the natural language techniques of text mining and information retrieval were created and refined during the boom of search engines and have only in the past few years begun to be used to mine the DDIs from large unstructured information bases, such as medical literature and EMRs. As a result, many new techniques have utilized modern NLP tools to map out DDIs and reduce the number of adverse drug events, including the three following methods.

Method A: Pharmspresso (Sentence Co-occurrence)

Garten, et al. [1] reports the use of text mining to automatically create a network of drug-gene relationships, using sentence-level co-occurrence in full-text of scientific articles. By doing so, they hope to score genes based on their likelihood of interacting with a specified drug. Although they do not directly search for DDIs, the network of drug-gene relationships can easily

be used to identify DDIs that occur as a result of two drugs metabolizing the same gene product, and done by a number of studies, including that of Percha, et al. [2], mentioned later.

The sentence-level co-occurrence methodology utilized by Garten, et al. improves on a previous algorithm by Hansen et al. called PGxPipeline [6], which requires two drug-gene knowledge bases and one protein-protein interaction network. These are namely: the Pharmacogenomics Knowledge Base (PharmGKB) and DrugBank (both for drug-gene interactions) and the InWeb interactome (for gene-gene interactions). Using these knowledge bases, PGxPipeline scores genes scores "based on their propensity to modulate drug response for a query drug" [1]. However, since the two drug-gene relationship databases PharmGKB and DrugBank are manually curated, they are expensive and difficult to maintain. Instead, Garten, et al. propose a new method that uses sentence-level co-occurrence in full-text medical literature using an algorithm that they previously developed called Pharmspresso to extract drug-gene relationships from the medical literature in place of the manually-curated databases. These new text-mined drug-gene relations are then applying the PGxPipeline to score these drug-gene relationships. The methodology used by Garten, et al. is mentioned below:

1. Obtain a corpus of full-text articles of drug-gene relationships. Garten, et al. used the QUOSA desktop application to automatically download the full-text PDFs from PharmGKB.
2. Generate a training set of drug-gene relationships. Garten, et al. extracted the relationships from the core tables of PharmGKB for all articles. For articles that contain more than one gene or more than one drug, relate all combinations of genes and drugs.
3. Run the Pharmspresso algorithm to create a drug gene network as follows [7]:
    a. Pharmspresso begins by initially tokenizing a corpus of full-text articles into sentences and words (Garten et al. do this using Perl scripts adapted from Textpresso search engine open source package. Note that Pharmspresso is built

on the Textpresso package with minor modifications for pharmacogenomic relationships.) [7].

   b. Then text is tagged in XML format with a tagging algorithm developed by Garten, et al. The text is tagged using a Pharmspresso ontology of two types of ontologies: biological entities (drugs, disease, genes, etc.) and relationships between entities (associations such as ‘binds’ and ‘interacts’, biological processes such as ‘acetylated’, ‘matures’, etc.). This tagging algorithm works using the Textpresso search engine’s templates, which themselves rely on a large written set of regular expressions to find templated relationships in text. The XML-tagged text is indexed for use by the Pharmspresso search engine.
   c. A drug-gene network is then created by drawing edges between genes and drugs that co-occur in the sentence level, with the weight of each edge corresponding to the number of articles supporting the relationship.

4. Run the PGxPipeline algorithm to score all genes according to their propensity for modulating drug response for a query drug as follows [6]:
   a. For a query drug and a given gene (referred to later as the original gene), the PGxPipeline first finds the gene’s direct interaction partners (first-degree connections in the network) in the InWeb gene-gene interaction network and links this gene to its partners.
   b. For each partner gene, note the drugs with which the partner gene interacts with in the drug-gene relationship network created by Pharmspresso.
   c. Using the drugs found in the previous step, score the ‘original gene’ based on the structural drug similarity to the query drug as defined as the Tanimoto coefficient of 166 structural features and based on the similarity of indications of use (encoded in Medical Entity Subject Heading terms) which also uses a Tanimoto coefficient [1]. The Tanimoto coefficient is defined in this case as the number of

features in common divided by the total number of features for both drugs [6]. Note that the original PGxPipeline used PharmGKB and DrugBank as sources of drug-gene interactions but Garten, et al. uses the drug-gene network created with Pharmspresso.

d. Train a logistic regression classifier on input positive examples (relationships derived in step 2) and three random negative examples (examples not in the positive set) using the features of structural similarity and similarity of indications (as defined as Tanimoto coefficients), mentioned in the previous step. Note that logistic regression classifiers, like all other machine learning classifiers, require positive and negative training examples in order to separate items (whose data on their features is own) into different classes. Hansen, et al. delve into greater detail about how they set up their logistic regression classifier in the supplementary data of their paper [6].
e. Use the logistic regression classifier to classify which drug-gene pairs are predicted to have a relationship with each other.

Ultimately, Garten, et al. compared their text-mining classifier against the original curation-based classifier using 5-fold cross validation on a gold standard set of drug-gene interactions. This gold standard consists a total of 682 unique drug-gene relationships from the 916 PharmKGB articles that mention at most one gene and one drug from. When using only known pharmacogenes (genes with some known relationship to a drug) as negative examples, with a only the 682 unique drug-gene relationships, Garten, et al. finds that “the text-mining-based classifier out-performs the original classifier, with (ROC) curves with area under the curve (AUC) of 0.701 and 0.672 respectively).” (Figure 1) [1]. This indicates a very low false positive rate, despite possible low recall. However, this result indicates that under some conditions, the text-mining-based approach can work just as well, if not better than the manually-curated

approach. Garten, et al. explains that this can be because identifying drug-gene relationships is not what "PharmKGB curators have been tasked with; they curate articles with respect to which genes and drugs that are mentioned without specifically asserting which genes and drugs relate. Therefore, it is not surprising that when using the high-quality gold standard that the text-mining classifier actually performs slightly better" [1].

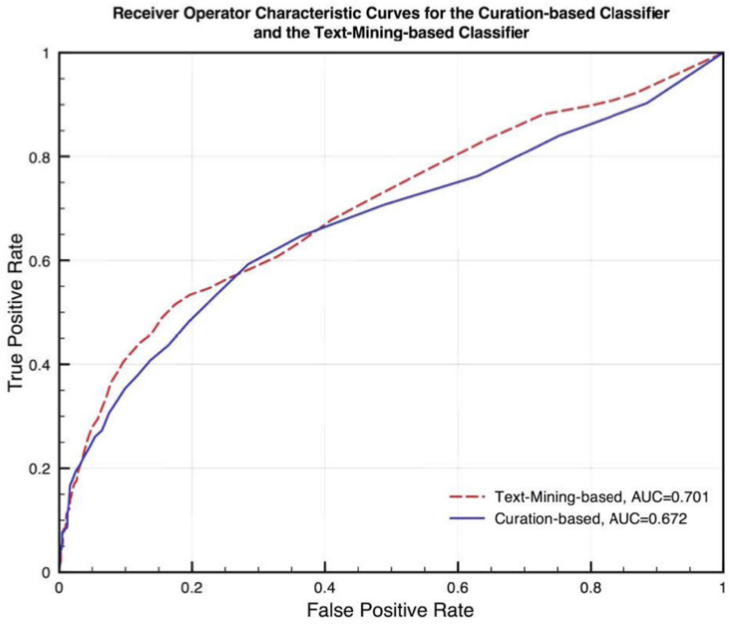

Figure 1 (from Garten, et al.). This shows the performance of Garten, et al.'s text-mining-based classifier against the curation-based classifier when using only known pharmacogenes as negative examples for the logistic regression classifier and a stricter gold standard for evaluation.

However, when using all genes as negative examples and expanding the definition of gold standard to all known drug-gene relationships in the PharmKGB data, Garten, et al.'s classifier generates an ROC curve with AUC value of 7.99 as opposed to 0.814.

In addition, Garten, et al.'s method can also be used to identify highly likely potential drug-gene relationships. They performed an external validation of their text-mining classifier by using 1,636 relationships that were added to PharmGKB after establishing the original training

set to score three times that many randomly chosen drug-gene relationships. Each relationship was scored leaving out knowledge of the relationship during training. In performing this external validation, Pharmspresso found many relationships that do not appear in PharmGKB but had high scores from the classifier and that co-occur in several sentences. These were marked as 'putative relationships' and submitted for review by PharmGBB, including the relationship between *CYP3A5* and cyclosporin, which has been verified and now has three articles in PharmGKB that support this relationship.

Since Garten, et al. create a classifier to determine drug-gene relationships, these relationships can easily be used to find instances of the subclass of DDIs where two drugs interact with the same gene. Since the combination of sentence co-occurrence in various articles combined with the PGxPipeline scores can help predict new drug-gene relationships, these new drug-gene relationships also facilitates the discovery of novel drug-drug relationships.

A great disadvantage to the use of sentence co-occurrence in medical literature is that this method cannot resolve cases where the explicit names of the two drugs are not in the same sentence. This can occur, for example, in the case of the sentence: "99% of all humans carry the same single nucleotide polymorphism at the gene G. This gene heavily affects the response to drug D." This is called anaphora, where an element of text refers to another element, and will not be picked up by simple sentence co-occurrence. As a result, there are bound to be a number of false negatives. There are a number of probabilistic models and computational methods being developed on this front, including DrugNerA, a NLP tool by Segura-Bedmar, et al. which performs sentence splitting, tokenization, POS-tagging, chunking, and linking of phrases with UMLS concepts to resolve anaphoras specifically related to biomedical literature [8].

However, the greater concern when using sentence co-occurrence to identify drug-gene relationships is the possibility of false positives, which are likely to occur if a drug and gene that have nothing to do with each other happen to occur in the same sentence. Since sentence co-occurrence can happen easily between unrelated drugs and genes in full-text research articles,

it is perhaps more likely for Pharmspresso to have false positives than false negatives. This trade-off between false positives and false negatives should always be taken into account when choosing a method for text-mining DDIs.

Furthermore, this method does take into account the type of relationship between the drug and the gene. Garten, et al. mention that it would be advantageous to know if a drug and gene have a positive or negative relationship or whether the gene is pharmacokinetic (gene or gene products affect drug) or pharmacodynamic (drug affects gene or gene products) for the drug. One way to improve on this is to use the existing tags to identify potential relationships between the drugs, since words of the sentence are already tagged with the Pharmspresso Ontology. One may be able to improve this method by experimenting with sentence co-occurrence of three elements: one drug, one gene and one relationship. The next paper by Percha, et al. proposes a solution to this using templates.

Method B: Template-based classifier

Percha, et al. present another gene-drug relationship classifier to infer novel DDIs through combining known facts expressed in scientific text. Unlike the previously mentioned sentence co-occurrence method, the method proposed by Percha et al. uses a random forest classifier to determine what types of gene-drug relationships best predict DDIs and also proposes a mechanism for the mechanism of interaction by assigning the type of interaction between each gene-drug pair (metabolize, inhibit, activate, etc.). The methodology is listed below:

1. Create two lexicons of terms: one for gene names and one for drug and drug-class names. Percha, et al. used a set of 731 known pharmacodynamic and pharmacokinetic genes and a set of 2,910 unique drug names (both from PharmGKB).
2. Obtain a corpus of article abstracts. They used all Medline extracts published before 2009.

3. Tokenize all articles into sentences and retrieve all sentences that mention both a drug and a gene of interest (hereafter, referred to as ‘seeds’).
4. Represent sentences as dependency graphs using a parser (in this case, the Stanford Parser). Dependency graphs are defined as rooted, oriented and labeled graphs, where words are nodes and edges are dependency relations between words. A sample dependency graph is given below in Figure 2. Discard sentences in which two seeds are not located in the same sentence and remove from sentences with more than one clause, clauses do not have both seeds.

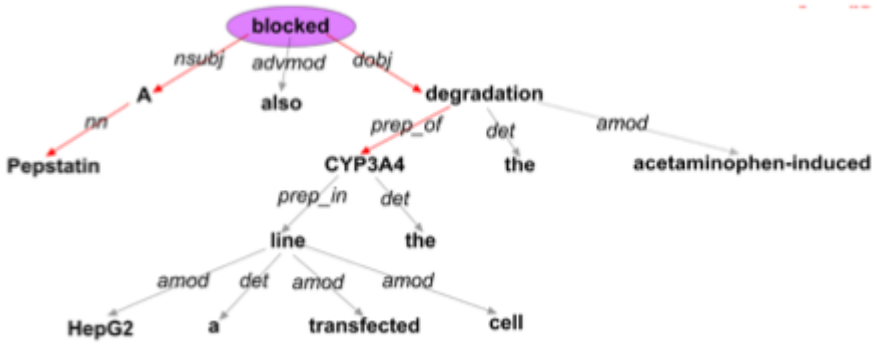

Figure 2 (from Percha, et al.). “This shows a sample dependency graph for the sentence ‘Pepstatin A also blocked the acetaminophen-induced degradataion of the CYP3A4 in a transfected HepG2 cell line”. The red arrow shows the path through the graph that connects the seeds Pepstatin A (a drug) and CYP3A4 (a gene). Because this path contains a verb - in this case, “blocked” - this is a sentence of interest” [2].

5. Normalize composite entities, such as ‘degradation of CYP3A4’ or ‘CYP3A4 elimination’ into the same concept (Elimination) using an ontology of concept terms.
6. Extract relations between composite entities. where relations are verbs (e.g. associated) or nominalized verbs (e.g. association) and composite entities are the two entities in a given sentence that are related by this verb.

7. Normalize relations (in the same way that context terms were normalized) to a set of normalized relationships taken from the ontology. For example, map ‘associated’ and ‘related’ to isAssociatedWith and map less common terms like ‘augment’ to ‘increase’.
8. Now with many normalized gene-drug relations of the form specified in Figure 3, put all relations into passive voice, collapsing passive/active pairs of normalized verbs (e.g. isMetabolizedBy and metabolizes) into single features and eliminate duplicate similar relations.

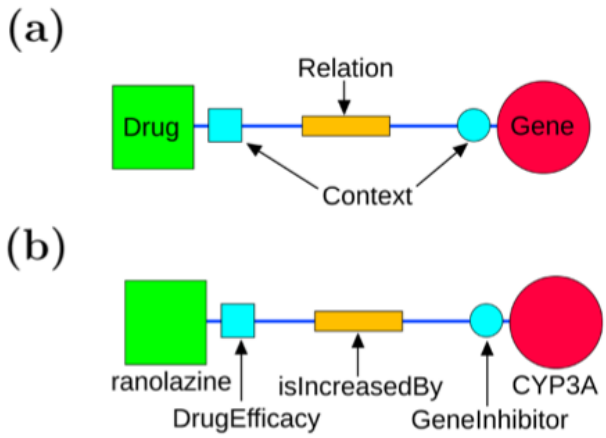

Figure 3 (from Percha, et al.). “A single drug-gene edge in the semantic network... (a) The general form of an edge. (b) A specified example.”

9. With your unique active-voice drug-gene relations, construct a semantic network, with each edge having the form listed in Figure 3.
10. Extract all the genes names, relation words, and context terms found on all of the shortest paths between two drugs to create your set of features. Find the shortest paths by performing a breadth-first search for every possible pair of drugs, D1 and D2. This assumes that the shortest textual path linking two drugs in the network represented the simplest explanation for the mechanism of their interaction. Note that in this paper, Percha, et al. made the decision to explore only paths of the form: D1 and context -

relation - gene and context - relation - D2 and context, because of computational feasibility and for ease of mechanistic explanation.

11. Train a random forest (a supervised machine learning classifier) to recognize interacting drug pairs based on the textual features of their connecting paths using 5000 known-to-interact drug pairs as the positive training examples and 5000 additional random drug pairs, with no overlaps with the positive set (in this case, Percha, et al. took all training pairs from DrugBank). Do this by recording all the features observed in every positive and negative training example. Percha, et al. recorded their features as unique numbers and used the R library implementation, randomForest to train and execute the random forest classifier.
    a. Note that random forests are ensemble methods in which many uncorrelated trees "vote" to classify data points and each tree uses only a subset of the features to maintain a lack of correlation. The number of trees must be set based on experimentation to minimize the overall error classification (Percha, et al. found that the classification error for their data set stabilized around 200 trees).
12. Run the classifier, using a voting cut of 50% of the votes of decision trees to classify a training point as positive. To determine the most likely mechanisms of interaction for a drug pairs, take the path with the highest number of decision tree 'yes' votes.

Percha, et al. evaluated the performance of their DDI classifier with the out-of-bag (OOB) error estimate, where they train each decision tree using only ⅔ of the available data. Note that there were a total of 172,271 negative training examples (paths between 5000 noninteracting drug pairs) and 182,534 positive training examples. Of the 172,271 negative training examples, it corrected classified 135,842 of them as non-interacting (78.85% specificity) and of the 182,534 positive training examples, it correctly classified 145,619 of them as 'interacting' (79.78% sensitivity). In total, the classifier correctly assigned 281,461 out of 354,805 training paths (and has an overall classification accuracy of 79.3%).

In addition, the random forest uses a permutation method to determine the most important textual features and were able to determine the 50 most important features, including but not limited to certain genes, context terms such as “Synthesis”, “Expression”, “DrugDose”, “DrugMetabolism”, “GeneInhibitor” and the relations “metabolizes”, “inhibits”, “suppresses”, among others.

Ultimately, once the random forest classifier has been trained, it is a very powerful tool for predicting new DDIs that are not yet known and a powerful tool for figuring out the mechanism by which a DDI occurs. Percha, et al. hope “to use the trained random forest to predict the most likely mechanisms of interaction between drug pairs that are often prescribed together but whose interaction status is not yet known.” [2].

Because classifier still operates on the sentence level, it is also susceptible to false negatives due to anaphoras, as acknowledged by Percha, et al. In future work, they hope to find ways to resolve anaphora, and they suggest “perhaps by considering pairs of entities that are mentioned in the same abstract, not just the same sentence.” Other modes of resolving anaphora are also being developed and can improve this algorithm, as the previously mentioned tool created by Segura-Bedmar, et al. [8].

However, this technique has a low probability of making false positives since the drug-gene relationship must match a certain template based on a limited ontology of 731 known pharmacodynamic and pharmacokinetic genes and a set of 2,910 unique drug names (both from PharmGKB) and 41 relation words. In contrast to the technique proposed by Garten, et al., it is far less likely to have a false positive if a drug and a gene exist in this structure in a sentence.

Furthermore, this technique also gives types of relations to drugs and genes, which was a limitation in the Garten, et al. study. This also enables it to propose multiple possible mechanisms of drug-drug interactions since we now have explicit drug-gene relations and ultimately gives an evaluation of certainty of each of these mechanisms (percentage of trees

that give ‘yes’ votes), which ultimately facilitates exploration of these potential pathways and verification of new DDIs. For example, in the classification of a known DDI between verapamil and atorvastatin, many paths between the two drugs go through ABCB1, where edges on one side of multiple paths indicate that verapamil inhibits ABCB1 activity and edges on the other side indicate that atorvastatin upregulates ABCB1 activity, indicating interfering functions. These paths all have greater than 90% of yes votes.

Method C: Using Text-Mining and Automated Reasoning

In addition to text-mining the medical literature, as did the previous two methods of mining DDIs from scientific literature, Tari, et al. [3] present a method that combines text-mining and automatic reasoning about biology to make high-quality predictions of new DDIs. Like Percha, et al., this approach also attempts to explain the mechanism of the interactions. The methodology used by Percha, et al. is as follows:

1. Performing an offline one-time parse of the 17 million Medline abstracts by extracting both explicit drug interactions and implicit drug interactions to turn the unstructured information into structural representation in a ‘parse tree database’. This is done using a number of pre-established external tools, including the Link Grammar parser [9] to parse the free text, BANNER to identify the official gene/protein names [10], MetaMap to identify the official drug name [11], and GNAT to disambiguate gene mentions [12] by identifying the official gene symbols for each gene mentions identified by BANNER. In the parse tree database, the structured information includes grammatical structures of sentences in the form of ‘parse trees’ (ordered, rooted trees that represent syntactic structure according to a language’s grammar, where nodes are part of speech tags and leaves are words) and the biological entities involved in the sentences. The parse tree database is managed by MySQL and the parse trees are accessed using PTQL, a language that Tari, et al. created to query the database.

2. Extraction of DDIs from the database is split into extracting explicit drug interactions, where a DDI is explicitly mentioned in a sentence, and implicit drug interactions.
    - To extract explicit drug interactions from sentences such as the following, **Enantioselective induction of 'cyclophosphamide' metabolism by 'phenytoin' (PMID: 10423284)**, one would use a PTQL query as such:

      ```
      //S{//?[Tag='Drug'](d1) =>
           //?[Value IN {'induce', 'induces', 'inhibit',
           'inhibits'}](v) => //?[Tag='Drug'](d2) =>
           //?[Value= 'metabolism'](w)} :: [d1 v d2 w]5 :
           d1.value, v.value, d2.value
      ```

      which specifies that a drug (d1) has to be followed by a form of 'induce' or 'inhibit' and then followed by another drug (d2) with the keyword 'metabolism'.
    - To extract implicit drug interactions, one must infer based on various properties of drug metabolism, in other words, through automatic reasoning. For example, one could extract <protein, metabolizes, drug> triplets with the following PTQL query:

      ```
      //S{/?[Tag='DRUG'](kw2) => //VP{//?[Value IN
           {'metabolised','metabolized'}](kw1) =>
                //?[Tag='GENE'](kw0)}} :::
                     kw0.value, kw1.value, kw2.value
      ```

      which specifies that a drug (kw2) is followed by verb phrase (VP) that includes a form of 'metabolized' and is followed by a gene mention (kw0).
3. In order to start reasoning, Tari, et al. transforms the extracted interactions for reasoning purpose by transforming them into their logic forms (e.g. the triplet <cyp3a4, metabolizes, lovastatin> is represented as extr(cyp3a4, metabolizes, lovastatin), protein(cyp3a4), drug(lovastatin), meaning that the 'cyp3a4 metabolizes lovastatin' relationship was extracted, cyp3a4 is a protein and lovastatin is a drug).
4. To ensure that the extracted relations are high-quality interactions and can be applied to derive DDIs, some data cleaning is done. The data cleaning involves incorporating general knowledge about drug metabolism and DDIs to filter out low-quality relations. To do this, Tari, et al. first identifies protein families and considers negative interactions.

- An example of identifying protein families is as follows:

  "A protein p is considered as an 'enzyme' if either one of the following holds (in the given order of precedence):
  - p belongs to the CYP, UGT or SULT gene families, i.e. official gene symbol starts with CYP, UGT or SULT;
  - p is annotated under UniProt as having keywords 'hydrolase', 'ligase', 'lyase' or 'transferase';
  - p is annotated under the GO term 'metabolic process' (GO: 0008152);
  - the Entrez Gene summary (provided by RefSeq) of p contains the key phrase 'drug metabolism' or the regular expression 'enzyme*' or 'catalyz*'." [3]

  If a protein p meets all of these requirements, then it is represented in the form of

  enzyme(p)

- An example of negative interactions is as follows:

  - In the following sentence: "... oxybutynin is predominantly metabolized by CYP3A4 and CYP3A5 but not by CYP2D6" (PMID: 9584328), negation words such as 'not' cause the negative interaction <CYP2D6, not_metabolizes, oxybutynin>.

Data cleaning is then done with AnsProlog [13] logic rules to refine interactions. AnsProlog is a declarative language that is useful for specifying what kind of reasoning to be performed rather than how to perform the reasoning. To clean the data, an AnsProlog logic rule would specify that in order to have the interaction metabolized(D, P), meaning that drug D metabolizes protein P, the following conditions must be fulfilled:

- D must be a drug
- P must be an enzyme (because only enzymes can metabolize substances)
- there must be no instances of extr(P, not_metabolizes, D).

After applying this AnsProlog logic rule, the extracted interactions extr(GENE1, metabolizes, oxybutynin) would turn into the logical facts metabolized(oxybutynin, GENE1) but extr(GENE2, metabolizes, oxybutynin) would not become a logical fact if extr(GENE2, not_metabolizes, oxybutynin) is among the interactions.

5. Tari, et al. then encode knowledge about drug metabolism in the form of AnsProlog logic rules and extract relationships to reason about potential DDIs and reveal the most promising ones. They do this by, for example, by making a rule where Dr1 decreases Dr2 if...
    - Dr1 increases the level of protein P
    - protein P is an enzyme
    - Dr2 is metabolized by P
    - Dr1 is a drug
    - Dr2 is a drug

    This is shown in the following AnsProlog logic rule given as an example in the Tari, et al. paper: `result(Dr1, decreases, Dr2) :- affects(Dr1, level(P, high)), enzyme(P), metabolized(Dr2, P), drug(Dr1), drug(Dr2).` In order to find the answer set of all drugs that match this AnsProlog logic rule, an AnsProlog solver called clingo [14] can be used to computer this set, which will be composed of DDIs among D.

Tari, et al. evaluated the performance of their method with a gold standard from DrugBank that was created by selecting 265 drugs, which had 494 known DDIs with the description that one drug increases or decreases the effect of another. They ran their algorithm on all Medline abstracts and found 170 predicted DDIs for the 265 aforementioned drugs, with 132 of the extracted interactions being true positives (77.7%), according to their originating sentences on DrugBank. However, only 20 of these true DDIs (11.8%) are annotated in the DrugBank standard, indicating that there is a potentially large number of published drug interactions that are not annotated.

Furthermore, Tari, et al. searched for novel induction/increase- and inhibition/decrease-based DDIs by running their method over all 17 million Medline abstracts and predicted 4154 direct DDIs of this type, where a drug directly increases or reduces an enzyme's levels, causing

a change in the levels of another drug that the enzyme metabolizes (see step 5 of the methodology section for an example). For direct enzyme inhibition or induction relationships, 108 predicted DDIs coincided with the DrugBank gold standard but while evaluating 315 unconfirmed interactions, they found that 256 of them (81.3%) have the correct evidence to support the existence of a DDI.

Tari, et al.'s method differs significantly from both Garten, et al.'s and Percha, et al.'s in that it uses many extracted drug-gene and gene-gene interactions to predict DDIs that are not explicitly stated in the text. The establishment of an automatic reasoning system that took into account biological knowledge is also unique to Tari, et al.'s approach.

As with any NLP information retrieval task, there is always the trade-off between sensitivity and specificity. Because of the stringent methodology used for data cleaning here, drug-gene relationships and explicit drug-drug interactions will only be treated as a logical fact if they pass a greater number of more difficult criteria. As such, a small set of very high-quality rules is established, which simultaneously discourages false positives and encourages false negatives.

Furthermore, the use of so many external tools for various steps of the method (such as the Link Grammar parser [9] to parse the free text, BANNER to identify the official gene/protein names [10], MetaMap to identify the official drug name [11], GNAT to disambiguate gene mentions [12], etc.) offers various chances for improvement in terms of building one's own tools customized to drug-drug interactions. For example, supplementing BANNER or MetaMap with lexicons of genes or solely pharmacogenes or drugs of interest can improve the method by customizing the parameters of your tools to the domain of your research, rather than using generalist tools. This, of course, should take care to avoid overfitting.

Conclusion

As mentioned in all three articles, the trade-off between false positives and false negatives is the problem of all classification tasks, and should always be considered when picking a classifier and setting the parameters and cutoffs. In order of presentation, the first method which used sentence co-occurrence throughout full-text articles (hereafter called Method A), was the most likely to have false positives and less likely to have false negatives while Percha, et al.'s technique, which used templates (hereafter called Method B) would have less false positives and more false negatives and the technique presented by Tari, et al., which incorporated biological knowledge and cleaned the data (hereafter called Method C), was least likely to have false positives and more likely to have false negatives because of the high standards that could rule out many potential DDI relationships. In the case of discovering new DDIs, it is more valuable to have fewer false positives since this reduces the noise in your DDI predictions and increases the selectivity and quality of your predictions. Since all DDIs need to be verified through experimentation, if your predicted DDI set has mostly true positives, then after investigation, many more DDIs will have been acknowledged and added to databases and increase the likelihood of preventing an adverse drug event involving a DDI. Therefore, the Tari, et al.'s method ranks the most favorable since it theoretically should produce the lowest number of false positives while predicting DDIs.

In terms of critically and constructively analyzing the methods, I believe that there is a lot for each method to take from the the others and that the best method will come from some combination of the three. Here are a few ways:

- Method A could be bettered with incorporation of the actual type of relations between genes and drugs and could use either of the methods used by Method B or Method C in order to do so.
- Method B could easily incorporate biological knowledge to make rules from Method C, such as the fact that only 'enzymes' can 'metabolize' drugs and the recognition of negative relationships.

- Both Methods B and C can try to incorporate drug structural and indication similarity from Method A as features for determining if two drugs may possibly be related or act on the same gene products.
- Method C could benefit from using the lexicons of known genes and drugs from Method B in addition to his own classification tools. This may reduce misclassification of genes or drugs in a sentence.

Natural language processing is still a relatively new field and the three methods presented above using sentence co-occurrence, template-based classification and automated reasoning with text-mining are some of the first attempt at discovering DDIs without manual curation. The number of articles on using text-mining on medical literature is still low and Garten, et al., Percha, et al. and Tari, et al. have 14, 11 and 45 citations, respectively (Google Scholar). The field still has much room to grow as many new techniques from Machine Learning and NLP are carried over to develop new techniques such as Segura-Bedmar's shallow linguistic kernel technique for DDI extraction [15] or as established techniques are refined and improved, such as the recent attempts to resolving anaphoras [16], which could really improve sentence-level techniques such as Garten, et al.'s and Percha, et al.'s. As new techniques develop, the ability to source DDIs (as well as other important information) from unstructured medical will improve, thus allowing medicine to better people's quality and longevity of life.

References

1. Garten, Y. A. E. L., NICHOLAS P. Tatonetti, and RUSS B. Altman. "Improving the prediction of pharmacogenes using text-derived drug-gene relationships."*Pac Symp Biocomput*. Vol. 305. 2010.
2. Percha, Bethany, Yael Garten, and RUSS B. Altman. "Discovery and explanation of drug-drug interactions via text mining." *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing*. NIH Public Access, 2012.

3. Tari, Luis, et al. "Discovering drug–drug interactions: a text-mining and reasoning approach based on properties of drug metabolism." *Bioinformatics*26.18 (2010): i547-i553.
4. Tatonetti, Nicholas P., Guy Haskin Fernald, and Russ B. Altman. "A novel signal detection algorithm for identifying hidden drug-drug interactions in adverse event reports." *Journal of the American Medical Informatics Association* 19.1 (2012): 79-85.
5. Olvey, E. L., S. Clauschee, and D. C. Malone. "Comparison of critical drug–drug interaction listings: the Department of Veterans Affairs medical system and standard reference compendia." *Clinical Pharmacology & Therapeutics*87.1 (2009): 48-51.
6. Hansen, Niclas Tue, Søren Brunak, and R. B. Altman. "Generating genome-scale candidate gene lists for pharmacogenomics." *Clinical Pharmacology & Therapeutics* 86.2 (2009): 183-189.
7. Garten, Yael, and Russ Altman. "Pharmspresso: a text mining tool for extraction of pharmacogenomic concepts and relationships from full text." *BMC bioinformatics* 10.Suppl 2 (2009): S6.
8. Segura-Bedmar, Isabel, et al. "Resolving anaphoras for the extraction of drug-drug interactions in pharmacological documents." *BMC bioinformatics* 11.Suppl 2 (2010): S1.
9. Sleator, Daniel DK, and Davy Temperley. "Parsing English with a link grammar." *arXiv preprint cmp-lg/9508004* (1995).
10. Leaman, Robert, and Graciela Gonzalez. "BANNER: an executable survey of advances in biomedical named entity recognition." *Pacific Symposium on Biocomputing*. Vol. 13. 2008.
11. Aronson, Alan R. "Metamap: Mapping text to the umls metathesaurus."*Bethesda, MD: NLM, NIH, DHHS* (2006).
12. Hakenberg, Jörg, et al. "Inter-species normalization of gene mentions with GNAT." *Bioinformatics* 24.16 (2008): i126-i132.

13. Gelfond, Michael, and Vladimir Lifschitz. "The stable model semantics for logic programming." *Proceedings of the 5th International Conference on Logic programming*. Vol. 161. 1988.
14. Gebser, Martin, Max Ostrowski, and Torsten Schaub. "Constraint answer set solving." *Logic Programming*. Springer Berlin Heidelberg, 2009. 235-249.
15. Segura-Bedmar, Isabel, Paloma Martinez, and Cesar de Pablo-Sánchez. "Using a shallow linguistic kernel for drug–drug interaction extraction." *Journal of biomedical informatics* 44.5 (2011): 789-804.
16. Segura-Bedmar, Isabel, et al. "Resolving anaphoras for the extraction of drug-drug interactions in pharmacological documents." *BMC bioinformatics* 11.Suppl 2 (2010): S1.