Extraction of CYP Chemical Interactions from Biomedical Literature Using Natural Language Processing Methods

Dazhi Jiao* and David J. Wild

Indiana University School of Informatics, Bloomington, Indiana, 47408

Received September 12, 2008

This paper proposes a system that automatically extracts CYP protein and chemical interactions from journal article abstracts, using natural language processing (NLP) and text mining methods. In our system, we employ a maximum entropy based learning method, using results from syntactic, semantic, and lexical analysis of texts. We first present our system architecture and then discuss the data set for training our machine learning based models and the methods in building components in our system, such as part of speech (POS) tagging, Named Entity Recognition (NER), dependency parsing, and relation extraction. An evaluation of the system is conducted at the end, yielding very promising results: The POS, dependency parsing, and NER components in our system have achieved a very high level of accuracy as measured by precision, ranging from 85.9% to 98.5%, and the precision and the recall of the interaction extraction component are 76.0% and 82.6%, and for the overall system are 68.4% and 72.2%, respectively.

INTRODUCTION

With the exponential growth of life science literature comes the increasing importance of automated information extraction technology. It has become ever more difficult for human researchers to keep track of the publications without the assistance of computers. There is an increasing need for researchers and life science database providers to automatically organize and retrieve information from publications. Many text mining systems have been and continue to be developed to meet these requirements.[1−3] In these systems, natural language processing (NLP) is playing a very important role in assisting effective information extraction.

In biomedical text mining systems, NLP techniques can fall into two broad categories: rule-based and statistics based. Rule-based methods use patterns matching to find wanted information from texts. One problem of the rule-based methods is the difficulty to compile a complete set of rules when the information extraction task becomes complex. Statistical NLP methods usually are based on machine learning methods and use results from analysis of different aspects of the sentence structure as features. In general, machine learning based methods are more suitable for complicated tasks. In order to use statistical NLP methods, a mandated requirement is that there must be a corpus (body of text) available for training and testing.[4]

In recent years, studies have been designed using both rule-based and machine learning based methods to automatically extract interactions between biological entities, for example, protein−protein interactions (PPI).[5−8] However, because of the difference between the syntactic expressions between biomedical entities and chemicals, and the nature of NLP systems to work well in one domain but not for another, it is not trivial to adapt these methods to the extraction of information concerning chemical entities. In comparison to the advances in biological text mining, very little work has been done in the field of chemistry text mining. For interaction extraction that involves chemical entities, the only work we are aware of is a rule-based method to extract protein and drug interactions.[9]

The system described in this paper is the starting point of our efforts to narrow the gap between text mining in chemistry and biology. It uses machine learning methods to identify cytochrome P450 (CYP), small molecules, and the interactions between them from the literature. The method depends on the results from syntactical and semantic analysis of texts using several common natural language processing techniques, including named entity recognition and dependency parsing as well as part of speech tagging. In the future, we are planning to improve our methods by incorporating more chemistry domain knowledge into our system as well capturing more chemical information, such as chemical structures.

There are two main reasons why we chose CYP and chemical interaction. The first is because of its importance in drug discovery and development and clinical applications.[10] CYPs can catalyze the oxidation of endogenous and exogenous compounds, and they are the major enzymes involved in drug metabolism. Induction or inhibition of CYP enzymes could affect the metabolism of some drugs, consequently increasing the plasma concentration, even possibly to a toxic level, and cause adverse reactions. For example, the combination of cerivastatin (CER) and gemfibrozil (GEM) has been reported to cause severe adverse effects including fatality, because of the inhibition of the CYP2C8-mediated metabolism of CER by GEM.[11]

The second reason is the availability of a data set. Reviews and online databases of CYP and chemical interactions have been published in recent years.[12,13] There are also freely available annotated data sets for training and testing NLP tools in CYP researches.[14]

* Corresponding author phone: (812) 856-0089; e-mail: djiao@indiana.edu.

10.1021/ci800332w CCC: $40.75
Published on Web 01/21/2009

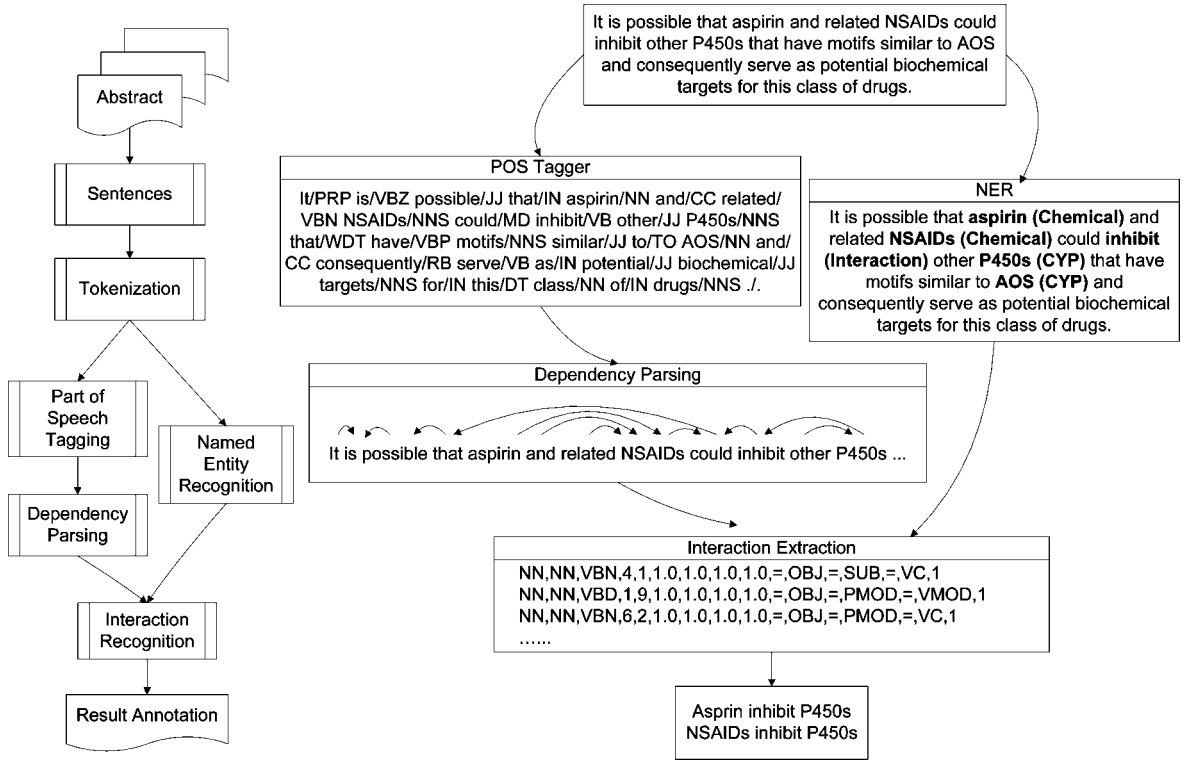

Figure 1. Left: The system architecture. Right: An example of how a sentence is analyzed step by step in the system to extract the interactions.

CORPUS

In natural language processing, a corpus refers to the gold standard used in training and evaluation of a system. There are many open accessible biomedical corpora from various sources.[14−17] The level of annotation differs in these corpora, from heavily annotated corpus that includes part of speech tagging (POS), constituent sentence structure, entities names, and relationships to simple annotated corpus that only contains POS tags and entities. For our system, we need a corpus that meets two requirements: it has to have rich chemical entities, and it needs to be heavily annotated with POS tagging, named entities, and syntactic structures so that it can be used for training different components in our system. Unfortunately, most of the corpora contain mainly biological terms such as proteins, genes, or even disease names. The only corpus that meets our requirements is the PennBioIE corpus. The corpus contains 2258 Medline abstracts in two domains: the molecular genetics of oncology, with 1158 abstracts, and the inhibition of enzymes of the CYP450 class, with 1100 abstracts. All abstracts are manually annotated for paragraphs, sentences, parts of speech, and biomedical named entities specific to each domain. A good portion of the abstracts are syntactically annotated in the Penn Treebank (PTB) style,[18] in other words, sentences in these abstracts are annotated in constituent trees.

The CYP450 abstracts in PennBioIE meet most of the requirements in our system, but it lacks annotated interactions. We manually annotated about 100 sentences with interactions between the annotated chemical and CYP entities. As research shows, annotation of corpus needs guidelines to keep the annotations consistent and achieve high quality.[19] In our case, guidelines proved to be very useful, especially when disambiguate occurs in extreme cases. We also developed annotation tools to facilitate the manual annotation process. Details of the tool can be found in the Supporting Information.

We also converted PTB constituent trees to dependency graphs that are required by the dependency parser component in our system, with the help of open source tools that convert constituent trees to dependency graphs.[20] We will discuss the differences between constituent trees and dependency graphs when introducing dependency parsers later.

SYSTEM ARCHITECTURE

In this section, we lay out the architecture of our system through a workflow of extracting interactions from an article abstract. We only briefly mention the technologies we use in the components of our system here, as these will be covered in the next section.

As illustrated in Figure 1 left side, abstracts are first broken into sentences. Each sentence is then tokenized, which means it is broken into its constituent tokens, or in this case, words. The tokens are then processed by two different processes: Named Entities Recognition (NER) and Part of Speech (POS) Tagging. The Named Entity Recognition component identifies the phrases or words that are of our interest—chemical, CYP, or interaction keywords. POS tagger assigns a part of speech tag to each token. The POS tags are the word classes based on its context and the definition belonging to a certain grammar, for example, in Penn Treebank, POS tag *NN* means *noun, singular, or mass*. The results of the POS tagger are then fed to a dependency parser for analysis based on a type of grammar called dependency grammar. The dependency grammar generates a graphical representation of the dependency structure of the sentence, which, together with the POS tags and named entities, are fed to the machine learning component, which identifies the interaction between a

chemical and CYP. All of the results, including tokens, POS tags, named entities, dependency graphs, and chemical CYP interactions, will be encapsulated in an XML file for future analysis. An example of how a sentence is decomposed and analyzed is given in Figure 1, right. We will discuss the details of this example in the following sections.

EXPERIMENTAL METHOD

As mentioned in the system architecture section, we use several natural language processing techniques in our system. Most of these techniques are based on a model that we built with our training data, the PennBioIE corpus plus our annotations of interactions. Here we discuss the methods we use in building these models in detail.

Tokenization. Tokenization is considered as a common task in text mining systems; however, here it is worth mentioning because of the complexity of patterns in our component caused by the syntax in chemical names. Common delimiters in English, such as the comma, are frequently used in chemical names. For example, common NLP tokenizers will break the chemical name *1,4-dihydropyridine* into several tokens: *"1", ",", "4", "-", "dihydropyridine"*. The tokenizer used in our system is a tokenization module in the OSCAR3 (Open Source Chemistry Analysis Routines), an open source chemical text mining application that can identify chemical terms and structures.[1] The OSCAR3 tokenizers can correctly keep the chemical name tokens, such as our above example, intact but, at the same time, break tokens such as *methylcholanthrene-treated* into three tokens "*methylcholanthrene*", *"-", and "treated"*.

POS Tagging. The POS tagger in our system, TnT (*Trigrams'n'Tags),* is a Hidden Markov Model (HMM) based tagger.[21] TnT uses trigram probabilities (the probability of a POS tag given the POS tags of the two words that appear before the POS tagged word in context) for the maximum likelihood estimate in HMM. TnT also uses a special smoothing method to solve the sparse data problem, in other words, it can perform accurately even when tokens in the testing set have not appeared in the training set. The smoothing algorithm of TnT makes it advantageous in processing chemical and biomedical literature, for the reason that no training set can ever cover even a small percentage of all names of chemicals and biological entity names.

TnT also builds a lexicon (a list of words in the corpus with POS tags) during the process. After building the model, we added over 66,000 compound names from ChEBI[22] to this lexicon and marked them as NN (noun). As shown in the evaluation section, this process in fact only slightly improved the accuracy of the POS tagger.

Named Entity Recognition. Various methods have been proposed in identifying named entities in biomedical or chemistry literature,[1−3,23,24] including dictionary based methods, pattern matching based methods, and machine learning based methods that identify entities using context. In our system, we used both the dictionary based method and the machine learning method.

The dictionary based NER method can achieve high accuracy and is suitable when the names are inclusive and pertaining to a small domain. Feng et al. have compiled a list of common CYP chemical interaction keywords from the literature based on frequencies of their appearances.[9] We extended the list by adding certain keywords from our own analysis of the corpus and used it as our dictionary to identify the interaction keywords in the text. The verbs of these words are listed in Table 1. In our system, we also used the nouns, adjectives, and difference tenses of these verbs. For example, for the word *inhibit*, we also identified *inhibitions, inhibition, inhibiting, inhibitors, inhibition, inhibited, inhibitory, inhibits, inhibit, inhibitive*, and *inhibitor* as interaction keywords. A full list of keywords can be found in the Supporting Information Table S1.

Table 1. List of Interaction Keywords (Verbs Only) Used for Identifying Interactions

list of interaction keywords		
accelerate	enhance	prevent
acetylate	expose	produce
activate	form	prohibit
affect	hydrolyse/hydrolyze	react
associate	improve	respond
bind	increase	stabilise/stabilize
block	induce	stimulate
carboxylate	inhibit	substrate
catalyse/catalyze	interact	suppress
control	interfere	transform
convert	ligand	
deacetylate	mediate	
decline	metabolise/metabolize	
decrease	modulate	
eliminate	oxidise/oxidize	

Table 2. IOB Labels Used in NER Component

IOB labels used in NER component		
B_CYP	B_SUBSTANCE	O
I_CYP	I_SUBSTANCE	

Example:
In/O addition/O, /O effects/O of/O each/O individual/O components/O of/O the/O reconstituted/O System/O, /O i.e./O, /O CYP1A1/B_CYP and/O P450/B_SUBSTANCE reductase/I_SUBSTANCE on/O 7-methoxyresorufin/B_SUBSTANCE O-demethylase/I_SUBSTANCE (/O MROD/B_SUBSTANCE)/O activity/O were/O studied/O./O

As mentioned above, OSCAR3 can analyze chemistry texts and identify chemical names and terms within. However, OSCAR3 does not always accurately identify biological entities, such as proteins and genes, in biomedical texts. Because the availability of the highly annotated corpus, PennBioIE, we decided to train our own NER model using the Stanford NER system, which is based on linear chain conditional random field (CRF) sequence models.[25]

In machine learning based methods, NER can be considered as a sequence labeling problem. The problem can be represented as the following: Given a list of n tokens $\{T_1, T_2, T_3, ..., T_n\}$ and a limited list of labels, find the corresponding sequence of labels $\{L_1, L_2, ..., L_n\}$. In our training process, we use the IOB labeling scheme that is adopted by many NER systems. Basically, B_ labels are used to mark the first token in an entity, and I_ labels are used to mark the rest of the tokens in the entity, and the remaining tokens are marked with O_ labels. The labels we used are listed in Table 2.

Dependency Parsing. Dependency parsers produce a directed graph (called a dependency graph) for input sentences which concerns only the dependency relationship between words and not with other issues, such as word order or structures such as phrases. In a dependency graph, each

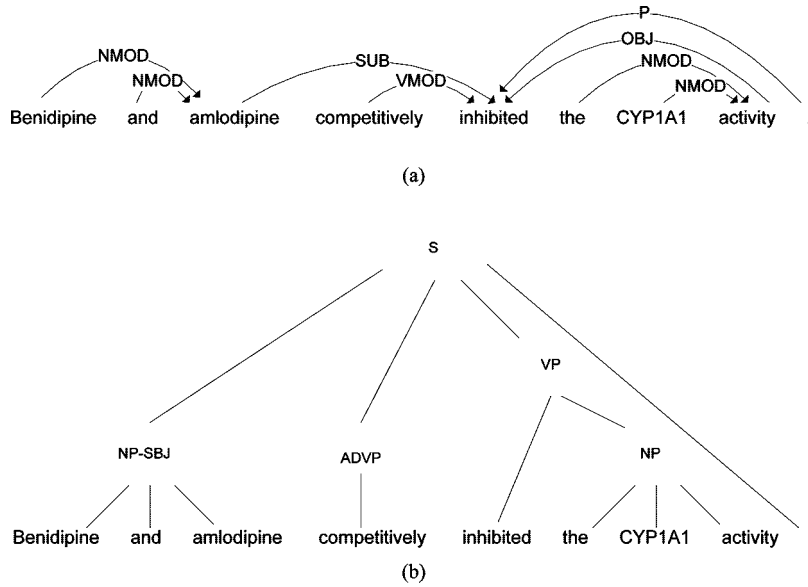

Figure 2. (a) A dependency graph. NMOD stands for noun modifier; VMOD stands for verb modifier; OBJ stands for object; SUB stands for subject; and P stands for punctuation. The root of the graph is the main verb (also called the head) of the sentence. (b) A constituent tree. The sentence node (S) is always at the root of the tree. NP-SBJ stands for noun phrase subject; ADVP stands for advert phrase; VP stands for verb phrase; and NP stands for noun phrase.

Table 3. List of Features Used in the Interaction Extraction

feature number	feature names
1	chemical POS
2	CYP POS
3	interaction POS
4	number of tokens between CYP and chemical
5	number of tokens between chemical and interaction
6	number of tokens between CYP and interaction
7	whether interaction is close to CYP and chemical
8	distance in dependency graph between CYP and interaction
9	distance in dependency graph between chemical and interaction
10	distance in dependency graph between CYP and chemical
11	whether there is a path from interaction to CYP in dependency graph
12	whether there is a path from interaction to Chemical in dependency graph
13	incoming arc label of chemical
14	Outgoing arc label of CYP
15	Outgoing arc label of interaction to chemical
16	outgoing arc label of interaction to CYP
17	CYP tokens
18	chemical tokens
19	interaction tokens

Table 4. Accuracy of POS and Dependency Parsing Components

	POS tagging	dependency parsing
training set	74647 tokens	74647 tokens 2943 sentences
testing set	8769 tokens	8769 tokens 316 sentences
accuracy (precision)	97.53%	98.5% (token level) 86.1% (sentence level)

node represents a word, and each directed arc represents a grammatical dependency such as the relationship between a verb and a noun with the noun being the subject (see Figure 2(a)). Constituent parsers are also often used in NLP systems. Constituent parsers recursively break the input text down into clauses and phrases and produce a tree structure (also known as parse trees or phrase structures) where the root represents the sentence as a whole, the leaves represent words, and the other nodes represent clauses and phrases (see Figure 2(b)). Dependency parsers have been widely adopted in biomedical text mining because dependency structures facilitate the extraction of entities or relationships, which most biomedical text mining systems require.[26,27] However, most annotated corpus are still in constituent structures (Treebank): for example, as mentioned above, the PennBioIE corpus is annotated in the Treebank formats (constituent structures), and we transformed it from its original format into dependency graphs. Recently corpuses that are annotated with dependency structures have started to become available; however, most of these corpuses are either automatically generated from the corpus[28] or automatically generated by a dependency parser.[16]

Our dependency parser is generated by MaltParser with the converted PennBioIE corpus. MaltParser is a data driven parser generator. It derives a dependency structure based on the syntactic analysis of a sentence and uses inductive machine learning methods to guide the parser at nondeterministic choice points.[29] The dependency parser took input from the results of POS tagger, and generates dependency graphs, that are used in our interaction extraction component.

Extraction of Interaction. Many works have been reported in the field of interaction extraction in biomedical field,[5−9,30,31] of which many focused on the interaction of proteins or genes. Interaction extraction can be considered as a relation extraction problem, and machine learning methods have been applied in biomedical text mining[5] and other domains such as news processing.[32] Relation extraction can be further viewed as a classification problem. For example, in our system, the extraction problem can be viewed as a problem to classify a combination of one chemical entity, one CYP protein, and an interaction entity as an interaction or noninteraction. In our system, we used the maximum entropy method to solve this classification problem. The maximum entropy method is a commonly used machine learning method in natural language processing tasks.[33] The

Table 5. Evaluation of the NER Component, Interaction Extraction Component, and the Whole System

	NER (CYP)	NER (chemical)	interaction extraction	whole system
training set	7112 entities	30957 entities	90 sentences (LOO) 189 interactions	
testing set	867 entities	4088 entities	90 sentences (LOO) 189 interactions	10 sentences 18 interactions
precision	85.9%	89.3%	76.0%	68.4%
recall	86.6%	89.2%	82.6%	72.2%
F-score	86.3%	89.3%	79.2%	70.2%

advantage of the maximum entropy method is its ability to include as much contextual information in the data as possible. It is based on the assumption that when in estimating the probabilities, the probability distribution that has the maximum entropy based on the data should be considered as the most likely distribution. The maximum entropy function is defined as in

$$P(o,h) = \frac{1}{\pi}\prod_{j=1}^{k}\alpha_j^{f_j(h,o)} \quad (1)$$

where o is the outcome, h is the feature vector, π is the normalization function, f_j is a feature function, and α_j is the jth model parameter. There is one model parameter and function for each feature. The parameters are determined by learning from the training data. The feature functions are binary functions. In the interaction extraction problem, the outcome can be only true or false. For example, one feature function for the interaction keyword "inhibit" (feature 18 in Table 3) can be defined as in

$$f(h,o) = \begin{cases} 1, \text{if } o = \text{true and interaction token} = \text{"inhibit"} \\ 0, \text{otherwise} \end{cases} \quad (2)$$

We use the opennlp maxent package in our system to train the maximum entropy model.[34] To help achieve accurate results, we explored different combination of features to cover syntactic, semantic, and lexical information. The list of features is listed in Table 3.

POS Features (1−3). The POS tags of the chemical and CYP are not as useful as the POS tag of the interactions, simply because the POS tags of the chemicals and CYP are mostly noun (NN, NNS) tags, while POS of interaction could be noun, verb, or adjective tags.

Distance Features (4−10). The number of tokens and the distance in the graph between entities are useful features because in general cases the interaction keyword and the chemical/CYP entities are close to each other in context, especially in the dependency graph. The distance in the graph is calculated as the sum of the distance from two nodes to their common ancestor or the distance between the two if the common ancestor is one of the two nodes. Feature 6 determines the relative position of the interaction entity is close to the CYP and chemical token. It is set to true if the interaction entity is between the CYP and chemical entity or if it is within 3 tokens from one of the chemical and CYP entities.

Relationship Features (11−12). These features identify whether there are dependencies between the interaction keywords and the CYP/chemical entities.

Label Features (13−16). As mentioned above, the dependency graph in our training set are converted from the Treebank format. The labels in the Treebank are also translated into labels used in the dependency graph. However, we can still consider the phrase structure being reserved in the dependency graph. This indicates that the constituent structure in the Treebank is also useful in our classifier. The reason could be that the entity names are normally a phrase (even though not represented in the dependency graph as a phrase), and thus the label which explains the syntactic category of the phrase can improve the classification model.

Token Features (17−19). These features include all the tokens that appear in the CYP or chemical names.

RESULTS

In this section we discuss the accuracy of each component in our system and also evaluate the performance of the whole system. In information retrieval and extraction systems, two popular measurements of the quality of the results are often used: precision and recall. Precision is the number of correctly extracted interaction divided by the total number of interactions extracted. Recall is the number of correctly extracted interactions divided by the total number of interactions in the testing set. In other words

$$Precision = \frac{TruePositives}{TruePositives + FalsePositives} \quad (3)$$

$$Recall = \frac{TruePositives}{TruePositives + FalseNegatives} \quad (4)$$

One combination of the two values is called F-score, which is another popular measurement in information retrieval and information extraction.

$$F = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (5)$$

To evaluate our system, we use recall, precision, and F-score to measure the performance of the NER component, the interaction extraction component, and the overall system. For components such as POS Tagging and dependency parsing, which can be considered as sequence label problems, we are only measuring the accuracy. This is actually equal to precision measurement, because we are measuring by the number of correctly labeled items over the total number of items. Since the number of tokens labeled in the result is the same as the testing set, the recall is always 100% and therefore not included in Table 4. The training and evaluation of POS Tagging and dependency parsing used the leave-many-out (LMO) method, where 90% of the random selected data are used for training and the rest of the data (10%) are left for evaluation. For the NER component, we also use the LMO method, but we calculate both recall and precision. For the interaction extraction component, due to the sparseness of the data, we decide to use the leave-one-out (LOO) method to evaluate it. We iterate our evaluation process 90 times, since the size of our training set is 90. During every iteration, one sentence is left out for testing using the maximum entropy model built from learning from the other 89 sentences. The recall and precision are calculated based

on the average results from these 90 iterations. The evaluation results of the POS and dependency parsing components are shown in Table 4.

We use 10 sentences for testing of the whole system. These sentences are not used in the training and evaluation of the interaction extraction component. We calculate the recall and precision of the interactions based on the results from these 10 sentences. We consider an interaction is correctly extracted only if the interaction keyword, the chemical entity, and the CYP entity are exactly the same as they are annotated in the testing set. We use precision and recall in the evaluation of the interaction extraction classifier and the overall system. The results are listed in Table 5.

Given the complexity of the system, the overall recall and precision are acceptable but can benefit from improvements. Besides, the precision and recall could be higher if we use a less strict method to count correct matches. For example, we noticed that in some extracted interactions from a sentence contains CYP entity as "cytochrome P450 (CYP450)", the original annotated CYP term is "cytochrome P450", but the extracted CYP entity is "CYP450". These are considered as incorrect matches in our current measurements.

We also notice that when testing the whole system using the 10 sentences, even though not all the interactions are correctly retrieved, all sentences are found to contain at least one interaction. This indicates that the system could be used as a preliminary filter for sentences or abstracts that contain targeted interactions between small molecules and CYPs.

CONCLUSIONS

Here we have presented a natural language processing system that can be used to extract chemical and CYP protein interactions from the literature with reasonable quality, based on machine learning methods in several components including NER, POS tagging, dependency parsing, and relation extraction. Such a system can be adopted to extract other types of interactions, such as chemical interactions with other types of proteins, because the models in our system can be trained using different corpus in other domains.

We are currently considering a number of improvements to the system. We expect to use a more advanced machine learning algorithm for training the interaction extraction model in the future. In addition, the model can be improved by annotating more training data. We also plan to add a component to convert chemical entities into structures, which will enable us to analyze the interactions on another level. In order to improve the accuracy of our methods, we also need to expand our annotated corpus. We will apply the findings in this research to extract other types of chemical information and biological information and integrate this system into the infrastructure of mining chemical and biological information that is currently being developed by our group at Indiana University School of Informatics.[35]

Supporting Information Available: Details of our annotation tool, example of annotated corpus, and our evaluation data set. This material is available free of charge via the Internet at http://pubs.acs.org.

REFERENCES AND NOTES

(1) Corbett, P.; Murray-Rust, P. High-throughput identification of chemistry in life science texts. *Computat. Life Sciences Ii, Proc.* **2006**, *4216*, 107–118.

(2) Batchelor, C.; Corbett, P. Semantic enrichment of journal articles using chemical named entity recognition. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*; Association for Computational Linguistics: 2007; pp 45−48.

(3) Furlong, L. I.; Dach, H.; Hofmann-Apitius, M.; Sanz, F. OSIRISv1.2: a named entity recognition system for sequence variants of genes in biomedical literature. *BMC Bioinf.* **2008**, *9*, 84.

(4) Manning, C. D.; Schütze, H. Introduction. In *Foundations of statistical natural language processing*; MIT Press: Cambridge, MA, 1999; pp 3−36.

(5) Xiao, J.; Su, J.; Zhou, G.; Tan, C. Protein-Protein Interaction Extraction: A Supervised Learning Approach. In *Proceedings of the First International Symposium on Semantic Mining in Biomedicine*; Hinxton, U.K., 2005.

(6) Temkin, J. M.; Gilder, M. R. Extraction of protein interaction information from unstructured text using a context-free grammar. *Bioinformatics* **2003**, *19* (16), 2046–53.

(7) Thomas, J.; Milward, D.; Ouzounis, C.; Pulman, S.; Carroll, M. Automatic extraction of protein interactions from scientific abstracts. *Pac. Symp. Biocomput.* **2000**, 541–52.

(8) Blaschke, C.; Andrade, M. A.; Ouzounis, C.; Valencia, A. Automatic extraction of biological information from scientific text: protein-protein interactions. *Proc. Int. Conf. Intell. Syst. Mol. Biol.* **1999**, 60–7.

(9) Feng, C. L.; Yamashita, F.; Hashida, M. Automated extraction of information from the literature on chemical-CYP3A4 interactions. *J. Chem. Inf. Model.* **2007**, *47* (6), 2449–2455.

(10) Huang, S. M.; Strong, J. M.; Zhang, L.; Reynolds, K. S.; Nallani, S.; Temple, R.; Abraham, S.; Habet, S. A.; Baweja, R. K.; Burckart, G. J.; Chung, S.; Colangelo, P.; Frucht, D.; Green, M. D.; Hepp, P.; Karnaukhova, E.; Ko, H. S.; Lee, J. I.; Marroum, P. J.; Norden, J. M.; Qiu, W.; Rahman, A.; Sobel, S.; Stifano, T.; Thummel, K.; Wei, X. X.; Yasuda, S.; Zheng, J. H.; Zhao, H.; Lesko, L. J. New era in drug interaction evaluation: US Food and Drug Administration update on CYP enzymes, transporters, and the guidance process. *J. Clin. Pharmacol.* **2008**, *48* (6), 662–70.

(11) Shitara, Y.; Hirano, M.; Sato, H.; Sugiyama, Y. Gemfibrozil and its glucuronide inhibit the organic anion transporting polypeptide 2 (OATP2/OATP1B1:SLC21A6)-mediated hepatic uptake and CYP2C8-mediated metabolism of cerivastatin: analysis of the mechanism of the clinically relevant drug-drug interaction between cerivastatin and gemfibrozil. *J. Pharmacol. Exp. Ther.* **2004**, *311* (1), 228–36.

(12) Rendic, S. Summary of information on human CYP enzymes: human P450 metabolism data. *Drug Metab. Rev.* **2002**, *34* (1−2), 83–448.

(13) Flockhart, D. *Drug Interactions: Cytochrome P450 Drug Interaction Table*, 2007 ed.; Indiana University School of Medicine: 2007.

(14) Kulick, S.; Bies, A.; Liberman, M.; Mandel, M.; McDonald, R.; Palmer, M.; Schein, A.; Ungar, L.; Winters, S.; White, P. Integrated Annotation for Biomedical Information Extraction. In *HLT-NAACL 2004 Workshop: BioLINK 2004, Linking Biological Literature, Ontologies and Databases, Association for Computational Linguistics*; Boston, MA, U.S.A., 2004; pp 61−68.

(15) Kim, J. D.; Ohta, T.; Tateisi, Y.; Tsujii, J. GENIA corpus--semantically annotated corpus for bio-textmining. *Bioinformatics* **2003**, *19 Suppl 1*, i180−2.

(16) Pyysalo, S.; Ginter, F.; Heimonen, J.; Bjorne, J.; Boberg, J.; Jarvinen, J.; Salakoski, T. BioInfer: a corpus for information extraction in the biomedical domain. *BMC Bioinf.* **2007**, *8*, 50.

(17) Tanabe, L.; Xie, N.; Thom, L. H.; Matten, W.; Wilbur, W. J. GENETAG: a tagged corpus for gene/protein named entity recognition. *BMC Bioinf.* **2005**, *6 Suppl 1*, S3.

(18) Marcus, M.; Santorini, B.; Marcinkiewicz, M. Building a Large Annotated Corpus of English: The Penn Treebank. *Comput. Linguistics* **1994**, *19* (2), 313–330.

(19) Kim, J. D.; Ohta, T.; Tsujii, J. Corpus annotation for mining biomedical events from literature. *BMC Bioinf.* **2008**, *9*, 10.

(20) Johansson, R.; Nugues, P. Extended Constituent-to-Dependency Conversion for English. In *Proceedings of NODALIDA 2007*; Tartu, Estonia, 2007.

(21) Brants, T. TnT - A Statistical Part-of-Speech Tagger. In *Proceedings of the Sixth Applied Natural Language Processing (ANLP-2000)*; Seattle, WA, 2000.

(22) Degtyarenko, K.; de Matos, P.; Ennis, M.; Hastings, J.; Zbinden, M.; McNaught, A.; Alcantara, R.; Darsow, M.; Guedj, M.; Ashburner, M. ChEBI: a database and ontology for chemical entities of biological interest. *Nucleic Acids. Res.* **2008**, *36* (Database issue), D344−50.

(23) Song, Y.; Kim, E.; Lee, G. G.; Yi, B. K. POSBIOTM-NER: a trainable biomedical named-entity recognition system. *Bioinformatics* **2005**, *21* (11), 2794–6.

(24) Lee, K. J.; Hwang, Y. S.; Kim, S.; Rim, H. C. Biomedical named entity recognition using two-phase model based on SVMs. *J. Biomed. Inf.* **2004**, *37* (6), 436–47.

(25) Finkel, J.; Grenager, T.; Manning, C. Incorporating non-local information into information extraction systems by Gibbs sampling, ACL '05. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*; Association for Computational Linguistics: 2005; pp 363−370.

(26) Clegg, A. B.; Shepherd, A. J. Benchmarking natural-language parsers for biological applications using dependency graphs. *BMC Bioinf.* **2007**, *8*, 24.

(27) Pyysalo, S.; Ginter, F.; Pahikkala, T.; Boberg, J.; Jarvinen, J.; Salakoski, T. Evaluation of two dependency parsers on biomedical corpus targeted at protein-protein interactions. *Int. J. Med. Inform* **2006**, *75* (6), 430–42.

(28) Schneider, G.; Rinaldi, F.; Kaljurand, K.; Hess, M. Steps towards a GENIA Dependency Treebank. In *Treebanks and Linguistic Theories (TLT) 2004*; Tübingen, Germany, 2004.

(29) Nivre, J.; Hall, J.; Nilsson, J.; Chanev, A.; Eryigit, G.; Kubler, S.; Marinov, S.; Marsi, E.. MaltParser: a language-independent system for data-driven dependency parsing. *Nat. Language Eng.* **2007**, *13*, 95–135.

(30) Domedel-Puig, N.; Wernisch, L. Applying GIFT, a Gene Interactions Finder in Text, to fly literature. *Bioinformatics* **2005**, *21* (17), 3582–3.

(31) Ono, T.; Hishigaki, H.; Tanigami, A.; Takagi, T. Automated extraction of information on protein-protein interactions from the biological literature. *Bioinformatics* **2001**, *17* (2), 155–61.

(32) Culotta, A.; Sorensen, J. Dependency tree kernels for relation extraction. In *Proceedings of the 42nd Annual Meeting of the Association of Computational Linguistics (ACL-2004)*; 2004; pp 423−429.

(33) Ratnaparkhi, A.; Brill, E.; Church, K. A Maximum Entropy Model for Part-of-Speech Tagging. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*; Association for Computational Linguistics: 1996; pp 133−142.

(34) Baldridge, J.; Morton, T.; Bierner, G. The opennlp.maxent Package. http://maxent.sourceforge.net/index.html (accessed 11/28/2008).

(35) Dong, X.; Gilbert, K. E.; Guha, R.; Heiland, R.; Kim, J.; Pierce, M. E.; Fox, G. C.; Wild, D. J. Web service infrastructure for chemoinformatics. *J. Chem. Inf. Model.* **2007**, *47* (4), 1303–1307.

CI800332W