Biomedical Information Extraction through Deep Parsing and Syntactic Role Matching

Anthony Gitter
Anthony.Gitter@asu.edu
Chitta Baral
chitta@asu.edu
Graciela Gonzalez
Graciela.Gonzalez@asu.edu

Department of Computer Science and Engineering, School of Computer and Informatics
Ira A. Fulton School of Engineering, Arizona State University

Abstract: The immense volume and scientific importance of protein-protein interaction information has created a need for an accurate biomedical information extraction system that can discover relevant interactions in the millions of biomedical journal abstracts and articles. Existing extraction systems often use hard-coded, rigid rules to detect such interactions. Our system, named Phoenix, stores the logic used to extract interactions in external rules than can be understood and customized by anyone with basic knowledge of English grammar. A novel, straightforward query language has been created to form these rules, which detect the syntactic roles of words or phrases in biomedical sentences that have been parsed into constituent trees.

1 Introduction

Reliable gene and protein interaction information is the basis for a wide range of biomedical research. Such interaction information has direct applications to drug design and the study of many serious and deadly diseases. For instance, the National Institute of Allergy and Infectious Diseases HIV Protein Interaction Project [1] states that understanding the protein-protein interactions between HIV and host cells is essential to preventing AIDS.

While an overwhelming number of protein-protein interactions have been discovered, such results are typically published in scientific journals using complex, technical natural language. PubMed [2], a widely-used biomedical information search tool, includes over 16 million citations and abstracts in its database. Although interaction information is abundant, it is often difficult for biomedical researchers to sift through this surplus of text to locate an interaction of interest. For instance, a PubMed search for the protein p53 yields approximately 40000 results. Clearly it is impossible for a researcher interested in p53 to read even the smallest portion of these abstracts or the accompanying full journal articles.

Fortunately, information extraction and natural language processing techniques can be used to pinpoint these interactions, store them in a database, and feed them into applications that can present interactions or interaction pathways cleanly to researchers. While a number of biomedical information extraction systems exist, our work aims to exceed the performance of current systems through its domain-independent, flexible architecture.

2 Related Work

Biomedical information extraction has gathered much interest in the past decade, in part because the complexity of biomedical language presents a compelling computational challenge and the benefits to be gained from a high-quality biomedical information extraction system are extensive. Initial protein-protein interaction extraction efforts employed simple co-occurrence methods [3], where an interaction was reported if two protein names were detected in the same sentence or abstract. Sometimes an interaction keyword like "*binds*" or "*activates*" was required as well to improve precision. However, nearly all biomedical information extraction systems now use some degree of natural language processing and syntactic parsing.

Syntactic parsers can be grouped into two broad categories: shallow and deep. Shallow parsers are fast because they perform only the most basic syntactic analysis such as part of speech tagging or phrase chunking. Typically, biomedical information extraction systems that incorporate a shallow parser [4-9] use some form of rules, patterns, templates, frames, or conditions to detect interactions after parsing, although unconventional approaches such as the graphical analysis of [9] exist as well. Systems that use deep parsing [10-17] not only examine the syntactic properties of individual words or phrases, but also determine the relationships between those words and phrases. Therefore, deep parsers provide more information than shallow parsers but are slower as a result. The system we present, Phoenix, uses deep parsing selectively to improve performance. In addition to a syntactic parse, biomedical information extraction systems may incorporate some level of semantic analysis. Moreover, machine learning may be used during extraction to learn the actual extraction rules and patterns [8, 16-18] or classify abstracts and sentences [19]. Cascaded finite state transducers and automata, as described by [20-22], have also proven to be an effective extraction technique.

While other biomedical information extraction systems have features in common with Phoenix, these existing systems are not as robust as Phoenix. Those such as [10] that also use the Link Grammar parser or a deep parse tree [12, 13] suffer from inflexible extraction algorithms that cannot be customized by users for specific tasks or improved as easily as Phoenix. While [4] uses a query language and the extraction technique in [5] allows user modification, these systems are limited because they only examine shallow parses. Thus, they cannot utilize the rich relationships between sentence components that are provided by a deep parse. In the following section we expand on these differences.

3 Approach

3.1 Constituent Parse Trees

A constituent is a single functional unit in a sentence such as a clause, noun phrase, or prepositional phrase. Link Grammar [23] is a deep syntactic parser that produces linkages, a syntactic structure in which pairs of words are connected with non-crossing links, as well as constituent parse trees. Essentially, constituent trees are a compact way to express the relationships between the clauses, phrases, and words of a sentence.

3.2 Extraction Query Language

Phoenix's extraction rules traverse a Link Grammar-produced constituent tree and detect the syntactic roles of the constituents in the sentence. Rules are written in a new, custom query language, which is partially syntactically derived from the LPath query language [24] and regular expressions. Incorporating these familiar operators helps users learn the query language quickly. A new query language was designed rather than using standard LPath in order to simplify rule construction and allow syntactic role labeling of tree nodes. Table 1 describes the query language's operators and their functions.

The query language has been designed so that it is possible to construct both general and specific rules as needed. Unlike the rules and patterns of some other extraction systems, Phoenix's rules are domain-independent. They do not rely on a protein name dictionary and are not specific to particular organisms or types of interactions. Furthermore, the plain text extraction rules can be stored outside of Phoenix's code so that the way Phoenix detects protein-protein interactions can be modified easily by users without programming experience. Anyone with knowledge of English grammar and an understanding of how rules may be constructed can adjust Phoenix without introducing bugs and without needing to comprehend how the extraction is implemented. Phoenix checks rule syntax so improper rules are rejected and do not cause errors. In addition, it is packaged with a set of default extraction rules so that users are not forced to write their own.

When applied to a constituent tree, each extraction rule can match a forest of diverse tree structures. The query language is efficient; in general, all rules traverse the tree in a single pre-order pass. During the traversal, the set of all rules begins at the root and expands and contracts as subsequent nodes are examined. When the current segment of a rule is matched to the current tree node, it is consumed and removed from the rule. Certain rules contain segments that may be matched multiple times, and copies of these rules are made before consumption. All rules that do not match the current tree node are removed from the set before the set is passed to all child nodes. Changes made to the set when

matching descendants of a particular child will not affect the set that is passed to the other children, each child has its own copy to manipulate. A tree node will only be visited multiple times when the current segment needs to examine the siblings of the current node. Such sibling rules interrupt the pre-order traversal by beginning a new pre-order traversal of each subtree whose root is a sibling of the current node. The initial set of rules for these traversals is composed of any matching sibling rules.

By noting the location of the "*SBAR*" constituents, which denote relative or embedded clauses in the tree, the subjects, verbs, and objects can by grouped by their source clause in the tree. Potential interactions are then formed by combining subjects, verbs, and objects in each clause, which creates triplets of the form <*subject*, *verb*, *object*>. Figure 1 above shows how the rules are used to detect two protein-protein interactions from the sentence "*c-Abl tyrosine kinase activity is blocked by pRb, which binds to the c-Abl kinase domain*" (from PMID 7828850).

4 Results

Early in Phoenix's development cycle we entered it in the international BioCreAtIvE II PPI-IPS (Protein-Protein Interaction task Interaction Pairs Subtask) [25]. The official results are found in Table 2. The scoring measures, precision, recall, and f-score, are defined below:

$$precision = \frac{truePositive}{truePositive + falsePositive} \quad \textbf{(1)}$$

$$recall = \frac{truePositive}{truePositive + falseNegative} \quad \textbf{(2)}$$

$$f - score = \frac{2 * precision * recall}{precision + recall} \quad \textbf{(3)}$$

where truePositive is the number of interactions extracted correctly, falsePositive is the number of interactions extracted incorrectly, and falseNegative is the number of interactions in the source text that were missed. Therefore, precision is the percent of extracted interactions that were extracted correctly, recall is the percent of interactions in the source text that were detected, and the f-score is the harmonic mean of precision and recall. While the BioCreAtIvE II PPI-IPS results are not fully indicative of Phoenix's performance potential due to its unfinished state at the time of submission, the challenge was an excellent opportunity to test the approach and compare it against other state-of-the-art biomedical information extraction systems. Phoenix was not one of the top performers, but its precision, recall, and f-score were within one standard deviation of the mean. Because roughly 68% of all data points fall within one standard deviation of the mean, we consider Phoenix to be an average performer rather than one of the worst entries. This suggests that refinements to the early implementation of our approach could boost Phoenix from an average biomedical information extraction system to one of the top systems. Furthermore, factors unrelated to the actual extraction algorithm, such as errors in gene and protein name normalization, severely reduced Phoenix's performance in the challenge. These issues are described in detail in [26]. Our internal testing, which like most extraction system evaluations did not require name normalization, shows Phoenix's performance to be significantly better than that reported in the BioCreAtIvE II PPI-IPS.

5 Future Work

Phoenix has shown great potential to capture a wide variety of protein-protein interactions, but there are many areas in which it can be improved and expanded. Even with perfect grammatical analysis, Phoenix cannot extract interactions expressed using anaphora. Anaphora resolution would allow for the extraction of interactions across multiple sentences and enhance extraction within a single sentence. In addition, pre-processing sentences before parsing them as well as examining all constituent trees returned by Link Grammar will reduce errors. Link Grammar often fails to return the best constituent tree representation first when parsing complex, biomedical sentences, and using the tree that most accurately expresses the grammatical relationships in a sentence will produce higher quality interactions.

Once the extraction rules detect subjects, verbs, and objects, more tree knowledge can be used to combine these components. We will modify the extraction algorithm to ensure that the subject, verb, and object are all part of the same

subtree and truly belong together. To improve the default set of rules, we can track which rules produce true positive and false positive interactions and adjust the rules accordingly. If we are able to annotate a large number of Link Grammar constituent trees, we would like to use machine learning to automatically learn a new default set of extraction rules.

Ultimately, we aim to move from protein-protein interactions to generic extraction of all relationships between biomedical entities expressed in the literature, such as gene-disease, gene-bio process, and gene-drug facts. This will require a much more flexible named entity recognition system, as the current system only tags gene and protein names. In addition, we will not be able to rely on protein-specific interaction keywords when filtering interactions. The verbs of interest will vary depending on the type of relationship being extracted.

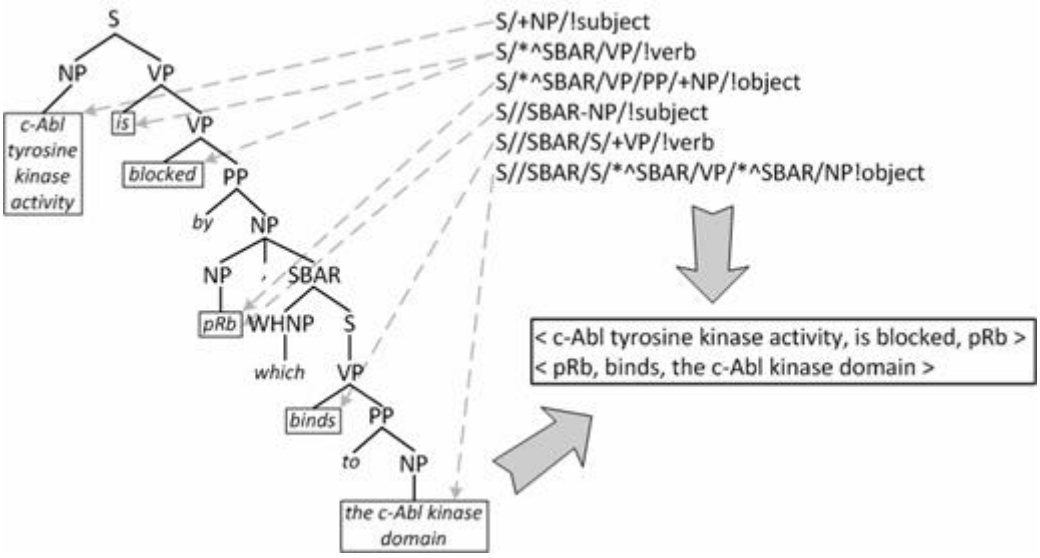

Figure 1. The set of all extraction rules applied to the constituent tree. Six of the rules match this tree's structure, identify the syntactic roles of leaf nodes, and generate two protein-protein interactions.

Table 1. The semantics of the rule transitions and operators of the query language. The capital letters in the examples denote constituent types. For instance, "*NP*" is a noun phrase and "*PP*" is a prepositional phrase.

Symbol	Meaning	Example	Translation
X/Y	Y is a direct child of X	S/NP	NP is a direct child of S
X//Y	Y is a descendant of X	S//PP	PP is a descendant of S
X-Y	Y is a sibling of X	SBAR-NP	NP is a sibling of SBAR
%	Node of any type	%/SBAR	SBAR is a child of any node
*X	Zero or more X nodes	*VP/NP	NP is the child of zero or more VP nodes, which are direct children of one another
+X	One or more X nodes	+VP/NP	NP is preceded by one or more VP nodes, which are direct children of one another
^X	Not an X node	^SBAR/VP	VP is a child of any node but SBAR
(X|Y)	Either X or Y	S/(VP|ADJP)	Either VP or ADJP is a child of S
!xyx	The syntactic role of this leaf node is xyz	NP/!subject	The leaf node that is a child of NP is a subject

Table 2. Official BioCreAtIvE II PPI-IPS results. Mean, Standard Deviation, and Median refer to all the entries submitted, and Phoenix's performance is shown in the rightmost column.

	Mean	Standard deviation	Median	Phoenix

Precision	0.0938	0.0881	0.0609	0.0343
Recall	0.1064	0.0704	0.1097	0.0717
F-score	0.0781	0.0505	0.0705	0.0464

References

[1] "NIAID HIV Protein Interaction Project." http://www.ncbi.nlm.nih.gov/RefSeq/HIVInteractions/

[2] "Entrez PubMed." http://www.pubmed.gov

[3] R. Jelier, G. Jenster, L. C. J. Dorssers, C. C. van der Eijk, E. M. van Mulligen, B. Mons, and J. A. Kors, "Co-occurrence based meta-analysis of scientific texts: retrieving biological relationships between genes," *Bioinformatics*, vol. 21, pp. 2049-2058, 2005.

[4] N. Domedel-Puig and L. Wernisch, "Applying GIFT, a Gene Interactions Finder in Text, to fly literature," *Bioinformatics*, vol. 21, pp. 3582-3583, 2005.

[5] D. P. A. Corney, B. F. Buxton, W. B. Langdon, and D. T. Jones, "BioRAT: extracting biological information from full-length papers," *Bioinformatics*, vol. 20, pp. 3206-3213, 2004.

[6] M. Huang, X. Zhu, and M. Li, "A Hybrid Method for Relation Extraction from Biomedical Literature," *International Journal of Medical Informatics*, vol. 75, pp. 443-455, 2006.

[7] C. Blaschke and A. Valencia, "The frame-based module of the SUISEKI information extraction system," *IEEE Intelligent Systems*, vol. 17, pp. 14-20, 2002.

[8] Y. Hao, X. Zhu, M. Huang, and M. Li, "Discovering patterns to extract protein-protein interactions from the literature: Part II," *Bioinformatics*, vol. 21, pp. 3294-3300, 2005.

[9] J. Cooper and A. Kershenbaum, "Discovery of protein-protein interactions using a combination of linguistic, statistical and graphical information," *BMC Bioinformatics*, vol. 6, pp. 143, 2005.

[10] S. T. Ahmed, D. Chidambaram, H. Davulcu, and C. Baral, "IntEx: A Syntactic Role Driven Protein-Protein Interaction Extractor for Bio-Medical Text. ," in *BioLINK SIG: Linking Literature, Information and Knowledge for Biology, a Joint Meeting of The ISMB BioLINK Special Interest Group on Text Data Mining and The ACL Workshop on Linking Biological Literature, Ontologies and Databases: Mining Biological Semantics (Biolink'2005)*. Detroit, Michigan, 2005.

[11] S. Novichkova, S. Egorov, and N. Daraselia, "MedScan, a natural language processing engine for MEDLINE abstracts," *Bioinformatics*, vol. 19, pp. 1699-1706, 2003.

[12] K. Fundel, R. Kuffner, and R. Zimmer, "RelEx - Relation extraction using dependency parse trees," *Bioinformatics*, vol. 23, pp. 365-371, 2007.

[13] H. Jang, J. Lim, J.-H. Lim, S.-J. Park, K.-C. Lee, and S.-H. Park, "Finding the evidence for protein-protein interactions from PubMed abstracts," *Bioinformatics*, vol. 22, pp. e220-226, 2006.

[14] C. Friedman, P. Kra, H. Yu, M. Krauthammer, and A. Rzhetsky, "GENIES: a natural-language processing system for the extraction of molecular pathways from journal articles," *Comput. Appl. Biosci.*, vol. 17, pp. S74-82, 2001.

[15] J. M. Temkin and M. R. Gilder, "Extraction of protein interaction information from unstructured text using a context-free grammar," *Bioinformatics*, vol. 19, pp. 2046-2053, 2003.

[16] T. M. Phuong, D. Lee, and K. H. Lee, "Learning Rules to Extract Protein Interactions from Biomedical Text," *PAKDD*, vol. 2003, pp. 148-158, 2003.

[17] H. W. Chun, Y. Tsuruoka, J. D. Kim, R. Shiba, N. Nagata, T. Hishiki, and J. Tsujii, "Extraction of gene-disease relations from MedLine using domain dictionaries and machine learning," *Proc. PSB 2006*, pp. 4–15, 2006.

[18] S. Katrenko, M. Marshall, M. Roos, and P. Adriaans, "Learning Biological Interactions from Medline Abstracts," *Learning Language in Logic Workshop (LLL'05) at ICML*, 2005.

[19] I. Donaldson, J. Martin, B. de Bruijn, C. Wolting, V. Lay, B. Tuekam, S. Zhang, B. Baskin, G. Bader, K. Michalickova, T. Pawson, and C. Hogue, "PreBIND and Textomy - mining the biomedical literature for protein-protein interactions using a support vector machine," *BMC Bioinformatics*, vol. 4, pp. 11, 2003.

[20] J. R. Hobbs, "Information extraction from biomedical text," *Journal of Biomedical Informatics*, vol. 35, pp. 260-264, 2002.

[21] J. Saric, L. J. Jensen, R. Ouzounova, I. Rojas, and P. Bork, "Extraction of regulatory gene/protein networks from Medline," *Bioinformatics*, pp. bti597, 2005.

[22] G. Leroy, D. M. McDonald, G. Ng, H. Chen, J. D. Martinez, S. Eggers, R. R. Falsey, K. L. Kislin, Z. Huang, and J. Li, "Genescene: biomedical text and data mining," *Proceedings of the third ACM/IEEE-CS joint conference on Digital libraries*, pp. 116-118, 2003.

[23] D. D. Sleator and D. Temperley, "Parsing English with a link grammar," *Third International Workshop on Parsing Technologies*, 1993.

[24] S. Bird, Y. Chen, S. Davidson, H. Lee, and Y. Zheng, "Extending XPath to Support Linguistic Queries," *Workshop on Programming Language Technologies for XML (PLAN-X)*, 2005.

[25] M. Krallinger, "BioCreAtIvE II - Protein-Protein Interaction Task," 2006.

[26] G. Gonzalez, L. Tari, A. Gitter, R. Leaman, S. Nikkila, R. Wendt, A. Zeigler, and C. Baral, "Integrating knowledge extracted from biomedical literature: normalization and evidence statements for interactions," presented at Proceedings of the Second BioCreative Challenge Evaluation Workshop, Madrid, Spain, 2007.