CONSTITUENT TREE-BASED EXTRACTION AND EVALUATION OF PROTEIN-PROTEIN INTERACTIONS FROM BIOMEDICAL LITERATURE

by

Anthony James Gitter

has been approved

March 2007

APPROVED (printed name, signature):

_____, _____, Director

_____, _____, Second Reader

_____, _____, Third Reader

Honors Thesis Committee

ACCEPTED:

_____

Dean, The Barrett Honors College

Abstract

We present two systems, Phoenix and BioEval, to address current limitations in biomedical information extraction (IE) and evaluation. Many existing protein-protein interaction (PPI) extraction systems are inflexible or applicable for only a certain subset of relationships. In addition, evaluation methods applied to such systems vary widely, severely hindering direct comparison of their abilities. Phoenix is a domain-independent biomedical IE system that uses easily modifiable rules to extract PPI. BioEval is an evaluation platform that facilitates the development of IE systems and the comparison of such systems.

1 Introduction

When developing, critiquing, or evaluating a biomedical information extraction, one must not lose sight of the end goal: providing accurate, relevant data to biomedical researchers. Recent scientific techniques have resulted in an explosion in the quantity of biomedical journal articles published. Unfortunately, this valuable information is only moderately accessible, as it is expressed in natural language, therefore making it difficult to search through and time-consuming to read and digest. In an attempt to meet this challenge and make it easier to utilize published biomedical information, the last decade has seen a number of innovative and widely varied approaches for automatically extracting biomedical information. Many extraction systems have had their share of success, but there remains a large gap between what can be extracted automatically and manually, leaving significant room for improvement. [1] suggests that this may be because (i) only a portion of information in text may be stated explicitly and the rest requires inference or

(ii) the majority of information can be captured by common cases, but the remaining facts become increasingly rare and increasingly difficult to extract.

The problem of biomedical IE still remains unsolved and complete biomedical natural language understanding is a distant goal, but even a partially accurate biomedical IE system can be of great use to researchers. When relationships (also called interactions or facts) between biomedical entities are extracted from text and stored in a database or other computer-workable form, they can be used directly or passed to other biomedical research aids. For example, the Collaborative Bio Curation (CBioC) system [2] uses an automatic biomedical IE system [3] to bootstrap a database with facts found in a particular abstract. It is then easier for a researcher to modify the extracted facts or add any missing facts than it is to read the entire abstract. Furthermore, once a number of researchers assert a given fact to be true, then other researchers can trust its correctness and do not need to read the abstract at all. Another system, PreBIND [4], automatically populates a database with PPI data, which can then be reviewed and submit to the manually curated BIND [5]. When applied to a real curation task, the system reduced the curation time by 176 days, 70% of its typical duration.

While biomedical IE systems have proven to be useful in real-world curation scenarios, selecting a single system for a particular task is another problem in itself. The methods used to evaluate biomedical IE systems are arguably more varied than the extraction methods themselves so it is not obvious which systems are the most successful and what exactly they are successful at extracting. Even if the domain is limited to the extraction of PPI, a biomedical IE system may only extract interactions of a certain type or between particular kinds of proteins. Furthermore, although precision and recall are

standard measures for reporting extraction success, there is no standard method for determining which facts are correct or how many facts exist in a given body of text. Thus, as noted by [6] and [7], a direct comparison of precision and recall is either ill-defined or not representative of the true differences between systems. Moreover, an IE system that is tuned for a particular task or domain is difficult to adapt to a different or more general task or domain. We present two systems, Phoenix and BioEval, to overcome these lofty challenges. Phoenix is a domain-independent biomedical IE system that uses easily modifiable rules written in a new query language to extract PPI. BioEval is an evaluation platform that facilitates the development of IE systems and the comparison of such systems.

2 Phoenix architecture

2.1 Overview

At a high level, Phoenix is a PPI extraction system that primarily processes abstracts accessible through PubMed[1], producing a list of PPI that is optionally normalized and ranked. Phoenix is not limited to PubMed abstracts, but this form of text is its most common source as it is freely available and easy to access.

Figure 1 depicts the core modules of Phoenix. PubMed abstracts are processed individually, one sentence at a time. A sentence is first cleaned by the Jericho HTML Parser[2] to replace HTML characters with their ASCII equivalents. ABNER [8] then tags all gene and protein names. Phoenix uses the ABNER model that has been trained on the BioCreAtIvE corpus, which does not distinguish between gene and protein names. These

[1] www.pubmed.gov
[2] http://sourceforge.net/projects/jerichohtml/

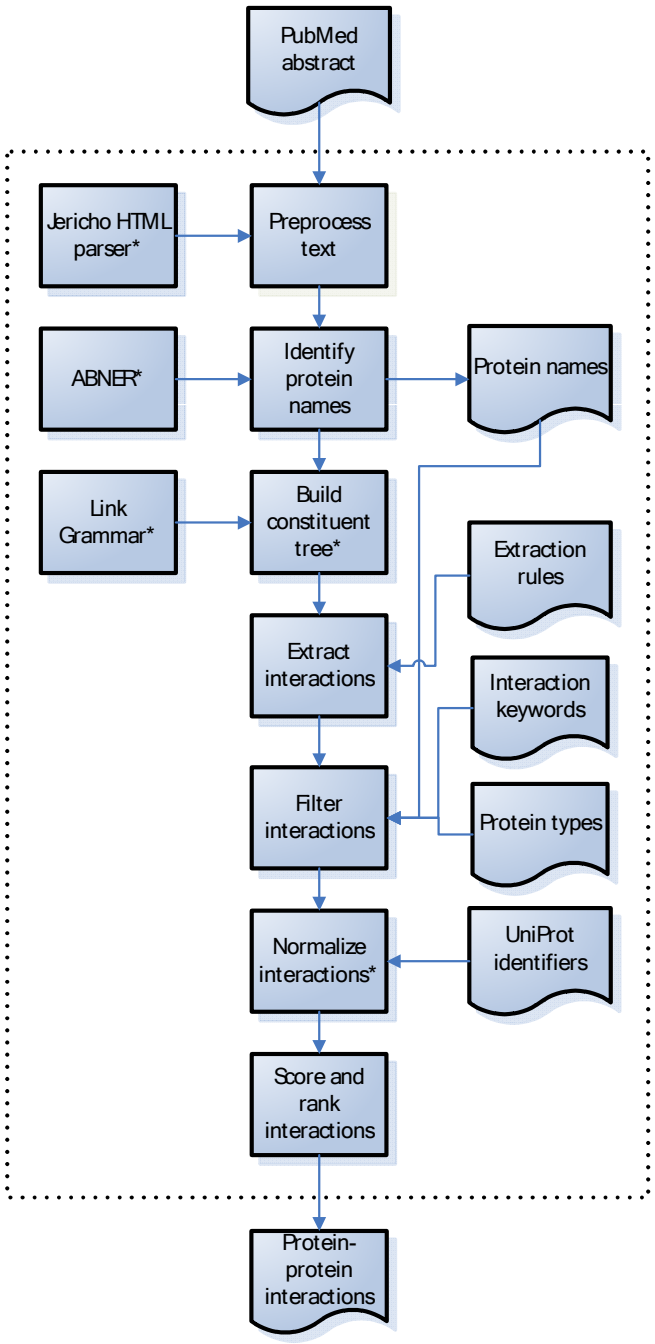


Figure 1. Phoenix's core modules. * denotes an external module created by the BioAI lab or a third party.

tagged protein names are also stored to be used by the interaction filter further downstream. Sentences that do not contain at least two protein names and an interaction keyword from the IEPA corpus list [9] are discarded.

The Link Grammar syntactic parser [10] then parses each sentence, producing a constituent tree (Section 2.2). The externally stored extraction rules are applied to the constituent tree, yielding a set of potential PPI (Section 2.3). These interactions are filtered so that only those interactions containing an interaction keyword and two of the protein names detected by ABNER remain. Each interaction is also cleaned by stripping articles and

trailing protein name types and generic words. For example, “the MITF gene” will be changed to simply “MITF”. This serves to aid the protein name normalization module, which relies on string similarity and is negatively affected by extraneous text.

Phoenix optionally normalizes protein names by mapping them to a list of UniProt [11] identifiers provided for the BioCreAtIvE II challenge Protein-Protein Interaction Task Protein Interaction Pairs Sub-Task (PPI-IPS) [12]. Normalization is performed using the Dice coefficient for similarity measurement. Details of the normalization procedure can be found in [13]. Each mapping is also assigned a confidence level.

If the extracted interactions are normalized, Phoenix can score and rank them (Section 2.4). Scoring is dependent on normalization because the score is partly based on how often a particular protein name and PPI occurs. Finally, interactions are written as an XML file. In addition to the interacting proteins and interaction keyword, other properties such as the source text, the location of the interaction (by paragraph and sentence), the rules used to detect the interaction, the interaction score, and the processing time are stored.

2.2 Use of the Link Grammar parser

Link Grammar 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 trees that follow the conventions of the Penn Treebank [14]. After analyzing a sentence, Link Grammar returns one or more linkages and/or constituent trees (as specified by the user)

depending on the ambiguity of the sentence. While not specifically designed for biomedical text, several biomedical IE systems [3, 15] employ it for syntactic parsing.

Interestingly, both [16] and [17] studied the use of Link Grammar for PPI extraction but came to different conclusions about its utility. [16] used the IEPA corpus [9] to compare PPI found through simple co-occurrence versus those that could be extracted using Link Grammar. They found that co-occurrence had 52% precision and 100% recall, whereas Link Grammar with a 10 minute timeout scored 61% precision and 87% recall, and therefore conclude Link Grammar is suitable for biomedical IE. On the other hand, [17] determine that Link Grammar in its standard form is not an appropriate parser in the biomedical domain. They construct a corpus by selecting pairs of interacting proteins from DIP [18] and use PubMed to find sentences that describe the interactions. With a 10 minute timeout, they report that it is possible to correctly extract a PPI from the first linkage returned 26.9% of the time and from the best linkage 58.1% of the time. As an alternative to the standard form of Link Grammar, they developed BioLG [19], a version of Link Grammar modified for the biomedical domain. Because BioLG is currently only available for Linux and Phoenix supports both Windows and Linux, Phoenix uses the standard Link Grammar parser instead of BioLG.

One limitation of Link Grammar in the biomedical domain is its inability to recognize biomedical entity names, which results in greater ambiguity and reduced parse accuracy. To partially remedy this problem, Phoenix appends the protein names ABNER detects with “GENE_”. Link Grammar assumes that unknown capital words are proper nouns so this ensures that protein names are properly recognized as proper nouns. In addition, spaces in multiword names are converted to underscores so that Link Grammar

does not split a single name into multiple constituents. Although [17] shows the first linkage is oftentimes not the best linkage, Phoenix does apply its extraction rules to the first constituent tree only. This is because it is difficult to automatically assess which tree (of potentially thousands of trees) is best. However, [20, 21] demonstrate that the Regularized Least-Squares algorithm and a Locality-Convolution kernel can be used to re-rank linkages in the biomedical domain for increased accuracy, but these techniques have not yet been incorporated into a Link Grammar or BioLG release.

2.3 Extraction query language

The extraction rules traverse a Link Grammar constituent tree and detect the syntactic roles of the constituent words or phrases in the sentence. Rules are written in a new, custom query language, which is partially syntactically derived from the LPath query language [22] 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. Figure 2 gives the grammar for Phoenix extraction rules. Each rule is composed of three main components. Rules begin with a RuleHead, which enforces the Link Grammar convention that all constituent parse trees begin with the "S" (main clause) constituent. One or more RuleBody sections give the syntactic structure that the rule will match, or in other words, the possible tree traversals that will result in a match. The RuleTail gives the syntactic role of the leaf node that has been matched. Table 1 provides the semantics of the various components of the query language.

ConsituentTreeRule ::= *RuleHead* [*RuleBody*] *FinalRuleBody RuleTail*
RuleHead ::= "**S**" *VerticalTransition*
RuleBody ::= *RuleBody RuleBody* | [*RepetitionChar*] ["**^**"] *BodyCore Transition*
FinalRuleBody ::= [*RepetitionChar*] ["**^**"] *BodyCore VerticalTransition*
RuleTail ::= "**!**" *SyntacticRole*
RepetitionChar ::= "*****" | "**+**"
BodyCore ::= "**%**" | *PartOfSpeech* | "**(**" *CompoundPartOfSpeech* "**)**"
CompoundPartOfSpeech ::= *PartOfSpeech* | *CompoundPartOfSpeech* "**|**" *PartOfSpeech*
PartOfSpeech ::= *AlphaChar* | *PartOfSpeech AlphaChar*
AlphaChar ::= **{all uppercase and lowercase characters in the English alphabet}**
Transition ::= *VerticalTransition* | *HorizontalTransition*
VerticalTransition ::= "**/**" | "**//**"
HorizontalTransition ::= "**-**"
SyntacticRole ::= *SyntacticChar* | *SyntacticRole SyntacticChar*
SyntacticChar ::= **{any character except "/" and "-"}**

Figure 2. A context-free grammar for Phoenix extraction rules. [ ] denotes optional terminals and nonterminals, *italics* denotes nonterminals, and **bold** denotes terminals. Because the rules can be recognized by a regular expression, there exists a regular grammar equivalent to this context-free grammar, but the context-free grammar is shown instead for readability.

Table 1. The semantics of the transitions and operations of the query language.

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

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 PPI 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 rules are implemented. Phoenix checks rule syntax so bad rules are rejected and do not cause it to crash. In addition, Phoenix 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. Figure 3 shows how a single extraction rule aligns with a multitude of constituent tree structures and is ultimately used to recognize two constituents as objects in a sample tree. The query language is efficient; generally, all rules traverse the tree in a single pre-order pass. The entire set of rules begins at the root, and the set 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. Because some rules contain segments that may be matched multiple times, copies of such rules are made before consumption. All rules that do not match the current tree node are removed from the set before it is passed to all child nodes.

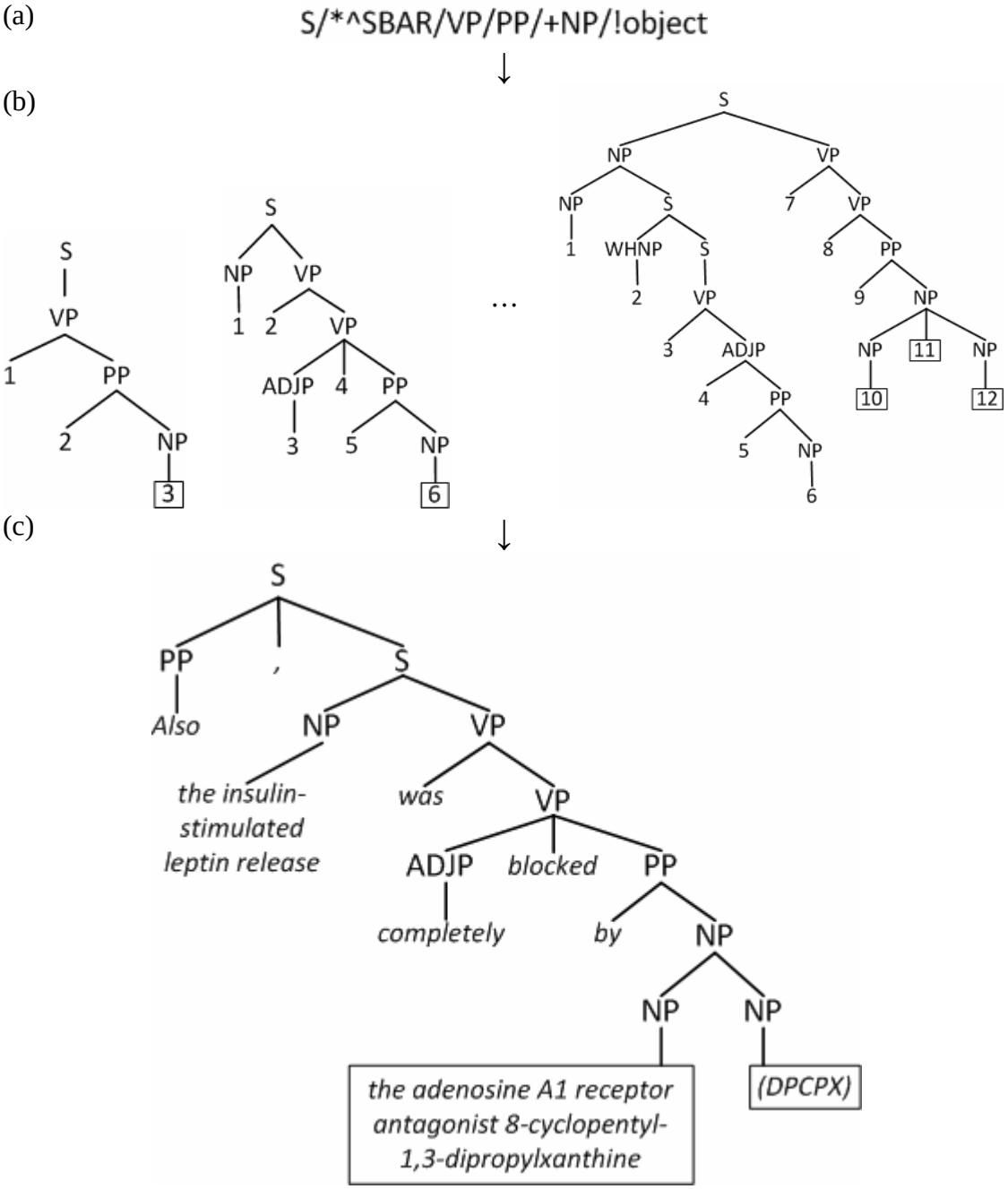


Figure 3. (a) An extraction rule to detect objects in a parsed sentence. (b) Three of the many possible tree structures that will be matched with the given rule. In these trees, numbers denote constituent words or phrases and boxed numbers indicate constituents that have been identified as objects. (c) In a sentence from PubMed identifier (PMID) 10615945, the rule detects "the adenosine A1 receptor antagonist 8-cyclopentyl-1,3-diproplyixanthine" and "(DPCPX)" as objects.

Changes made to the set of rules when matching descendants of the left child will not affect the set passed to the right child or other children. A node will only be visited multiple times when a rule that examines the siblings of the current node is present. 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" (embedded or relative clause) constituents 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 4 shows the full extraction of PPI from the sentence "c-Abl tyrosine kinase activity is blocked by pRb, which binds to the c-Abl kinase domain" (PMID 7828850).

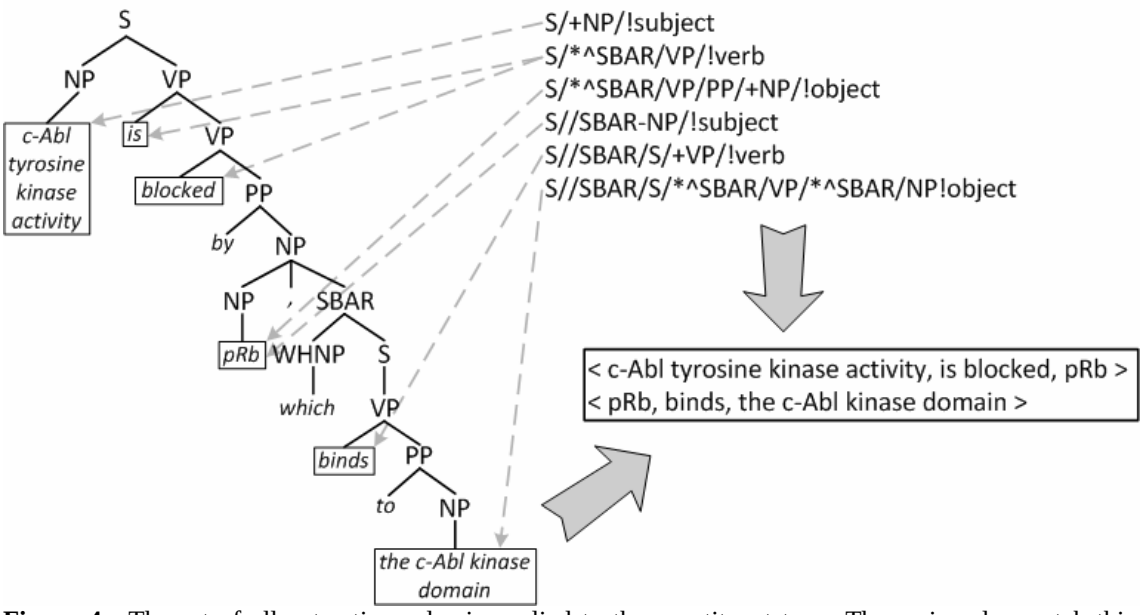


Figure 4. The set of all extraction rules is applied to the constituent tree. These six rules match this tree's structure, identify the syntactic roles of leaf nodes, and generate two protein-protein interactions.

2.4 Interaction scoring and ranking

Phoenix optionally scores PPI to assert the confidence in each interaction. Before scoring, any interaction in which a protein interacts with itself is removed, as are multiple occurrences of an interaction within a single sentence. All interactions begin with a base score of 1.0, which is the modified in the following manner:

- Increment the score by 2.0 if it was extracted from the abstract or conclusion[3].
- Increment the score by 1.0 if it was extracted from the results section.
- If an interaction occurs independently in two sentences, increase the score of the first occurrence by the score of the second. Remove the second occurrence, or set its score to 0 depending on the settings.
- Increment the score by a weight relative to how frequently its interacting proteins appear in the entire abstract or article.
- Increment the score by the average of the normalization confidence levels from the normalization step.

Once the interactions have been scored, they are sorted in descending order.

If a user chooses to normalize, score, and rank the PPI Phoenix extracts, score thresholds can be set to determine which interactions are ultimately output. When using score thresholds, with each pass through a body of text, Phoenix will output two sets of interactions: a high precision set and a high recall set. The high precision set corresponds to the higher score threshold so that only high confidence interactions are output. The high recall set corresponds to the lower score threshold. Thus, recall is likely to increase because more interactions are reported, but precision may drop as more incorrect interactions may be output as well.

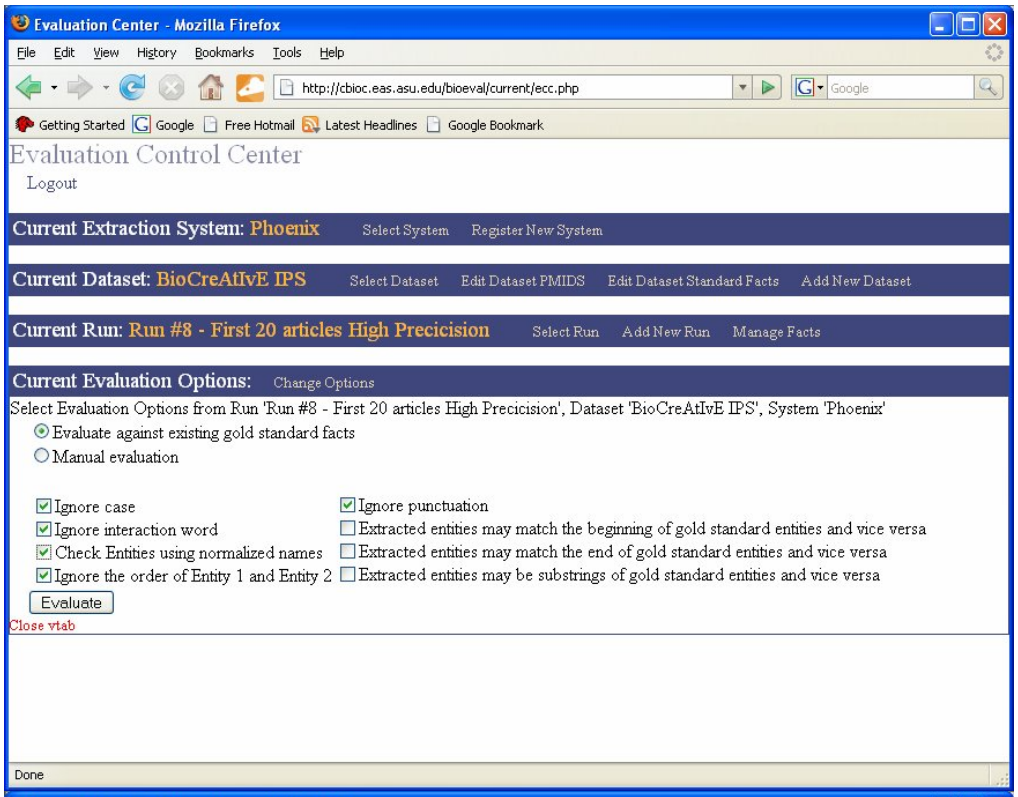
[3] The scoring steps related to the conclusion and results section are only applicable when extracting from full text articles.

3 BioEval architecture

3.1 Overview

BioEval can be used to evaluate runs from an extraction system, share gold standards, or directly compare biomedical IE systems. To begin evaluating extracted PPI, users must first select the extraction system that generated the set of interactions (the run) and the collection of documents that was processed (the dataset or corpus). There are a number of ways to build a dataset, including automatically gathering a set of PMIDs through a PubMed query or entering a list of PMIDs manually. Each corpus requires a set of gold standard facts which are stored in XML and uploaded to BioEval. Once a corpus and gold standard are prepared, the user uploads one or more runs to be scored. Finally, the evaluation options are selected. These options are shown in Figure 5(a) and include the evaluation measure to be used in evaluation (Section 3.2). By changing the options, the user has great control over the strictness of the evaluation and the resulting score. While these choices strongly effect evaluation results and must be made whenever evaluating PPI, typically they are not reported. Making the user choose the options explicitly ensures that they are given full attention and their effects can be accounted for when comparing evaluations.

(a)



(b)

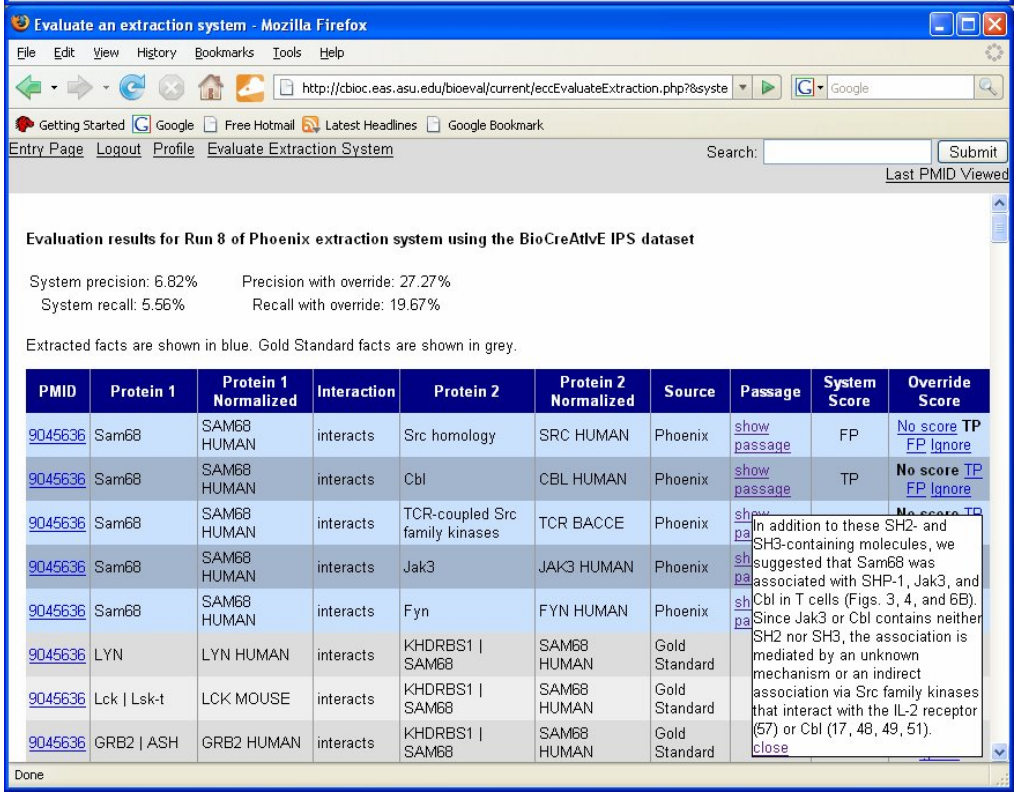


Figure 5. (a) After selecting the Extraction System, Dataset, and Run, the user chooses the Evaluation Options. (b) BioEval automatically calculates the precision and recall of a run using the specified options. The user may also override the automatic score of any individual fact based on a manual comparison of the extracted fact, the gold standard facts, and the source passage. The depicted interfaces have been improved from their original implementation by members of the BioAI lab.

As shown in Figure 5(b), BioEval calculates the precision and recall for the evaluated run. Precision (Equation 1) is the ratio of correctly extracted facts to all extracted facts. Recall (Equation 2) is the ratio of correctly extracted facts to all facts in the source text. In the following equations, let TP be the number of true positives or interactions extracted correctly, FP be the number of false positives or interactions extracted incorrectly, and FN be the number of false negatives or interactions in the text that were missed.

Equation 1. Precision calculation

$$precision = \frac{TP}{TP + FP}$$

Equation 2. Recall calculation

$$recall = \frac{TP}{TP + FN}$$

BioEval also displays all extracted facts in the run alongside all gold standard facts grouped by PMID. If the user includes all possible interaction details when uploading the run, each extracted fact shows the interacting proteins, their normalized forms, the interaction keyword, the source passage, and how BioEval scored the fact. Extracted facts are scored as TP, FP, or ignored. All copies of an interaction besides the first will be ignored if the interaction was extracted multiple times from the same PMID. Gold standard facts that do not match extracted facts are scored as FN or ignored.

If a user disagrees with the automatic scoring of extracted facts or gold standard facts, he or she may manually override the score. Precision and recall with override are calculated to reflect the changes, and the original system scores are shown as well for reference. Score overrides may be necessary, for instance, if a user is evaluating against an incomplete gold standard and determines that a FP does appear in the source passage as a legitimate interaction.

3.2 Evaluation measures

Evaluation measures are generally discussed in the context of named entity recognition evaluation, but they are equally important in evaluation of PPI extraction. One way to determine if a PPI is correct is to pair each extracted fact with each gold standard fact and individually assess the protein names and optionally the interaction keyword. Interaction keywords can be compared using stemming, but a lack of standardized gene and protein nomenclature [23] complicates entity evaluation. Exact string matching is overly strict, because a single name may have multiple acceptable forms, e.g., the gold standard for BioCreAtIvE Task 1A [24] included alternate forms for each gene name mention. For this reason, a number of less strict evaluation measures have been proposed for named entity recognition evaluation. [25-29] present the following as alternate evaluation measures:

- *Left match*: the extracted entity may be a left substring of the gold standard entity
- *Right match*: the extracted entity may be a right substring of the gold standard entity
- *Left/right match*: either a left match or right match is accepted
- *Partial match*: some substring of the extracted entity matches a gold standard entity
- *Approximate match*: the extracted entity is a substring of the gold standard entity or vice-versa
- *Fragment match*: each token of an extracted entity is evaluated individually against each token of gold standard entities
- *Sloppy match*: any part of the extracted entity may match any part of a gold standard entity
- *Protein name parts match*: a sloppy match where each token is considered individually
- *Core-term match*: a core-term is identified for each entity, and these core terms are used when matching against the gold standard

BioEval supports a subset of these evaluation measures through its evaluation options. Users may choose from exact, left, right, left/right, or approximate matching. Like approximate matching, left and right matching is bidirectional in BioEval. Supporting

several less strict evaluation measures allows users to evaluate the quality of and difficulties faced by the named entity recognition component of their IE system.

4 Results

4.1 Phoenix results

Early in Phoenix's development cycle, it was entered into the BioCreAtIvE II PPI-IPS [12]. While the results are not indicative of Phoenix's full performance potential due to its unfinished state, the challenge was an excellent opportunity to test the approach and compare it against other state-of-the-art biomedical IE systems. The official results can be found in Table 2.

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

Precision and recall are defined as above. F-score (Equation 3) is the harmonic mean between precision and recall.

Equation 3. F-score calculation

$$f-score = \frac{2 * precision * recall}{precision + recall}$$

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.

4.2 BioEval results

As an evaluation platform, BioEval cannot be quantitatively assessed in a straightforward manner like Phoenix. Its utility was confirmed while preparing Phoenix for the BioCreAtIvE II PPI-IPS. By loading the supplied training data into BioEval as a gold standard and building a dataset around that gold standard, we greatly reduced the time it took to examine the effects of changes to Phoenix and identify weaknesses. BioCreAtIvE II PPI-IPS used exact string matching of the normalized protein names and ignored the interaction keyword. Using BioEval, we could quickly view the source passage for all facts scored FP to determine if incorrect normalization, a lack of experimental evidence, or an error in the extraction algorithm caused the incorrect extraction. It is difficult to estimate how much development time was saved by using BioEval, but we can report that several distinct bugs were revealed that would have been time consuming to uncover otherwise.

5 Discussion

5.1 BioCreAtIvE II PPI-IPS versus typical evaluations

It is important to note that the BioCreAtIvE II PPI-IPS was more difficult than typical biomedical IE tasks. Each protein interaction pair submitted had to be supported by experimental evidence in the source article. Therefore, if an extracted interaction was mentioned in the source article but was supported by a reference to another paper instead of direct evidence, it would be scored as a false positive. In addition, each protein name was to be mapped to a UniProt [11] identifier. String-based mapping was insufficient because many identifiers were nearly identical with only the associated organism differing. These two challenges almost certainly led to the very low mean scores (significantly less than what is reported in literature) and impaired Phoenix as well.

Furthermore, additional unofficial evaluations of Phoenix still resulted in f-scores lower than those reported by state-of-the-art extraction systems. We believe this to be caused in part by the different aims of Phoenix and related systems. Phoenix is designed to be useful in a wide range of real-word biomedical research tasks. It intends to detect any type of reported interaction between any two genes or proteins. On the other hand, a review of related work (Section 6.1) reveals that many authors restrict the universe of potential interactions, sometimes so much that the resulting systems cannot be used for general extraction. For instance, authors may only consider interactions that

- Contain a particular interaction keyword (such as “bind” or “interact”)
- Contain proteins from a particular organism
- Contain protein names from a predefined list or lexicon
- Express a positive relationship between the entities

In all cases, the restrictions simplify the extraction task, thereby enhancing the resulting scores. Moreover, authors do not always state the test corpus and sometimes select a corpus so small that it is not representative of biomedical text as a whole.

5.2 Analysis of BioCreAtIvE II PPI-IPS results

Phoenix heavily relied on ABNER's protein name mentions for both sentence classification and triplet filtering. Using the model trained on the BioCreAtIvE corpus, ABNER reports 65.9% recall. Therefore, assuming independence of protein name recognition and ignoring the possibility that a false positive is identified, there is a 56.6% (Equation 4) chance that a sentence with exactly two protein names will be ignored because both names are not recognized.

Equation 4. Probability that a sentence of interest will be ignored because of ABNER

$$ignore = 100\% - (65.9\% * 65.9\%) = 56.6\%$$

In addition, a single false positive from ABNER can cause multiple false positives in the extracted interaction pairs if the incorrect protein name is present in multiple interaction pairs.

Additional errors were traced to Link Grammar and the rules used to extract interaction pairs from its constituent tree output. At the time of submission, Link Grammar split multiword protein names when building a constituent tree. This made normalization of the interaction pairs much more difficult, but has since been corrected. Moreover, Link Grammar produces many possible linkages and constituent trees for each sentence, but the first linkage and constituent tree returned by Link Grammar was always used for the extraction. Upon manual examination, it was found that the first linkage and

tree returned were not always the best representation of the sentence structure. In addition, much of the information to be gained by using a deep parse instead of a shallow part of speech (POS) tagging was not utilized. In Phoenix, subjects, verb phrases, and objects were grouped into sets and combined based on the clause of the sentence that contained them, rather than the tree structure. The rules themselves covered only the most general sentence constructs, which led Phoenix to overlook protein interactions expressed in less common grammatical forms.

Nearly all of the remaining errors were due to normalization. Organism disambiguation was based purely on which organisms appeared most often in the training data, which was an overly simplistic approach. For example, in one case, a correctly extracted interaction pair was normalized to human proteins instead of yeast proteins, even though this article's title alone, "The Cap-binding protein eIF4E promotes folding of a functional domain of yeast translation initiation factor eIF4G1", shows that IF4E and IF4G1 should be mapped to yeast proteins. Thus, contextual clues need to be examined when selecting the correct organism.

5.3. Phoenix future work

Phoenix has shown great potential to capture PPI, 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 for one or both protein names. Anaphora resolution would allow for the extraction of interactions across multiple sentences and enhance extraction within a single sentence. The methods proposed by [20, 21] can be incorporated into Phoenix to provide a higher quality constituent tree for the extraction

rules. As noted previously, Link Grammar struggles 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 better potential PPI.

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. In addition, we will create rules to handle nominalized interaction keywords. 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 set of extraction rules.

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

5.4 BioEval future work

BioEval is a wonderful aid in extraction system development and comparison at the moment, but there are many features that will make it more powerful and easier to use. We envision comparison of multiple systems at the same time – presently the user must

evaluate each system individually and then compare the results. Such a feature would also enable side-by-side comparison of different runs from the same extraction system. In addition, we will expand BioEval so that it can evaluate relationships other than PPI. This will eventually include evaluating normalization of protein names.

6 Related work

6.1 Biomedical IE systems

Biomedical IE has gathered much interest in the past decade, in part because the complexity of biomedical language presents a very difficult challenge the benefits to be gained from a high-quality biomedical information extraction system are extensive. Challenges such as those presented in [12, 30-33] spurred many innovative approaches using a wide variety of techniques. Initial PPI extraction efforts employed simple co-occurrence methods where an interaction was reported if two protein names were detected in the same sentence or abstract, possibly only in the presence of an interaction keyword. Modern, more sophisticated co-occurrence-based extraction systems include [34], which uses co-occurrence as a source for the associative concept space and distance to measure relatedness of concepts in the space. However, nearly all biomedical IE 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 quick because they only perform high level syntactic analysis such as POS tagging or phrase chunking. Typically, biomedical IE systems that incorporate a shallow parser [35-43] use some form of rules, patterns, templates, frames, or conditions to detect interactions after parsing, although unconventional approaches such as the

graphical analysis of [38] exist as well. Deep parsers 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. Biomedical IE systems that rely on a deep parser include [3, 6, 15, 16, 44-50] and Phoenix itself. In addition to a syntactic parse, biomedical IE systems may incorporate some level of semantic analysis. Moreover, machine learning may be used during extraction to classify abstracts and sentences [4] or learn the actual extraction rules and patterns [15, 43, 50, 51]. Cascaded finite state transducers and automata, as described by [1, 52, 53], have also proven to be an effective extraction technique.

Of all the existing biomedical IE systems, a few are of particular interest because of their similarity to Phoenix. IntEx [3] also uses the Link Grammar parser to detect the syntactic roles of words and construct PPI. Rather than use the constituent tree representation, IntEx deals directly with linkages. At its core, it uses a series of conditions that examine the links between words to detect and combine syntactic roles. Both RelEx [44] and the system described by [46] use deep parsers to build parse trees and extract relationships via manually written rules. The key difference between these three systems and Phoenix is the flexibility provided by Phoenix's query language. The other systems are rigid and cannot be adapted to accommodate additional patterns as easily as Phoenix. Similar customizability is also present in BioRAT [36], which allows users to define their own extraction templates. However, because BioRAT performs a shallow parse instead of a deep parse, the templates are very specific and cannot incorporate direct relational information like Phoenix. GIFT [35] is another related

shallow parsing biomedical IE system. It uses the CQP query language [54] to find PPI, but like BioRAT is only able to examine POS tags. This prevents GIFT queries from utilizing the grammatical relationships provided by a deep parser as Phoenix does.

6.2 Extraction system evaluation

An examination of the methods used to evaluate biomedical IE systems reveals the dire need for a standard evaluation platform. To elaborate on the discussion in Section 5.1, the corpus used for evaluation may oftentimes be too small to be truly representative of the extraction system's ability on biomedical text in general. One extreme example comes from [48], where the impressive 96% precision and 63% recall reported is for a corpus consisting of a single article. Other systems reporting high precision and recall may take a limited view of what constitutes a PPI. [39] reports 94.3% precision and 86.8% recall, but they consider only protein names appearing in a yeast protein name dictionary and four interaction keywords: interact, associate, bind, and complex. Such a narrow view of PPI significantly reduces the complexity of the PPI extraction task.

Many systems, such as those described in [6, 15, 37-40, 43, 50, 52], are evaluated using custom corpora that consist of PubMed abstracts or collections of sentences from biomedical text that have been manually annotated. Unfortunately, these corpora are rarely shared so the resulting scores cannot be directly compared due to differences in annotation styles and corpus complexity. In order to further legitimize the corpus used for evaluation, some systems [3, 36, 46] use established public PPI databases such as DIP [18] to construct a gold standard. However, such databases often have strict curation guidelines and contain interactions from full-text articles instead of only abstracts.

Therefore, the gold standard interactions obtained may not coincide well with the interactions that exist in a corpus composed of PubMed abstracts.

6.3 Automatic evaluation and challenges

There have been some attempts to automate or semi-automate the process of biomedical information extraction evaluation. The LLL Genic Interaction Extraction Challenge [30] provided a web interface for participants to upload their extractions. A Perl program was used to automatically score the entries and report the results. Similarly, the BioCreAtIvE II PPI-IPS distributed a Python evaluation script with its training set data. Both evaluation programs considered only exact matches as true positives and required some form of standardized names: canonical forms in a provided dictionary for LLL and UniProt identifiers for BioCreAtIvE II. These and other similar basic text-matching evaluation tools provide advantages over completely manual evaluation, but are not as rich as BioEval. They do not allow users to share datasets, use different evaluation measures, adjust the calculated scores, or store evaluation results online.

In the first BioCreAtIvE challenge [31], an online evaluation tool was created for task 2 to aid Gene Ontology Annotation (GOA) curators evaluate submissions. The tool displayed evidence and surrounding text supplied by the participants and highlighted the evidence text. The GOA curators then used this text to judge if the Gene Ontology annotation or prediction was correct. Furthermore, [55] present data and a procedure for automatically generate test suites for NER systems. They allow the length, case, numeric features, punctuation, presence of Greek letters, and other features of the entities in generated source sentences to be varied. The evaluation is performed using exact match,

but the authors note that users can post-process output or use their own code to generate the test suites with the data if they wish to employ other evaluation methods.

Information extraction challenges have proved to be one effective way to discover the state-of-the-art strategies in the field by evaluating many systems' performances on a given task. [56] discusses the evolution of the Message Understanding Conferences, which paved the way for modern biomedical information extraction challenges. The Knowledge Discovery and Data Mining Challenge Cup [32] asked participants to examine journal articles and determine which papers were most likely to need curation, whether each paper should be curated, and if experimental evidence existed for the products of each gene in the paper. The Bio-Entity Recognition Task at JNLPBA [33] required teams to recognize protein, DNA, RNA, cell line, and cell type names from MEDLINE abstracts. The BioCreAtIvE [31] and BioCreAtIvE II [12] tasks span a variety of challenges in biomedical information extraction including identification of gene mentions in abstracts, gene name normalization, annotation of gene products with Gene Ontology terms, and extraction of PPI. The LLL Genic Interaction Extraction Challenge evaluated the ability to learn rules that identify interactions between genes/proteins and their roles in the interaction.

While results of such challenges are of great value, they come at a great expense. Creating a training and test data set and evaluating submissions is typically done manually by curators and other experts and is a labor intensive process. For instance, Knowledge Discovery and Data Mining Challenge Cup coordinators approximate that their staff and FlyBase[4] curators spent nearly 11 staff months preparing and running the competition. Like these challenges, BioEval requires a test corpus that must be manually

[4] http://flybase.bio.indiana.edu/

constructed, but once the corpus is complete there is extremely little additional labor necessary to evaluate multiple IE systems. Moreover, the challenges provide a snapshot of the best system at the time of the evaluation, but cannot account for post-submission improvements or new systems that did not participate in the challenge. BioEval provides the same standard evaluation with a common test set and evaluation options, but is repeatable and does not require judges (although challenge judges could use it to support official challenge evaluations).

7 Conclusion

Taken together, Phoenix and BioEval aim to solve different aspects of the biomedical IE problem. Through its dynamic protein-name recognition and constituent tree query language, Phoenix is a flexible PPI extraction system that is not tied to a particular corpus or type of interactions. It is unique in its customizability and provides straightforward access to the rich information of a constituent tree with its extraction rules. On the other hand, BioEval is a platform that can standardize the evaluation of biomedical IE systems. By making typically unstated assumptions and evaluation parameters explicit and hosting shared corpora and gold standards, it is finally possible to intelligently make direct comparisons of different biomedical IE systems.

8 Acknowledgements

Very special thanks to Dr. Chitta Baral and Dr. Graciela Gonzalez. Without their ideas, support, and guidance this work would not have been possible. Thank you to Dr. Yi Chen who also served on the thesis committee. Thanks as well to the BioAI lab's

BioCreAtIvE team – Bob Leaman, Shawn Nikkila, Luis Tari, Craig Teegarden, Ryan Wendt, and Amanda Zeigler – all of whom directly contributed to critical components of Phoenix and/or BioEval. This work was funded in part by the Fulton Undergraduate Research Initiative.

References

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

[2] C. Baral, H. Davulcu, M. Nakamura, P. Singh, L. Tari, and L. Yu, "Collaborative Curation of Data from Bio-medical Texts and Abstracts and its integration," in *Data Integration in the Life Sciences*, vol. 3615, *Lecture Notes in Computer Science*, 2005, pp. 309-312.

[3] 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.

[4] 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.

[5] G. D. Bader, I. Donaldson, C. Wolting, B. F. F. Ouellette, T. Pawson, and C. W. V. Hogue, "BIND--The Biomolecular Interaction Network Database," *Nucl. Acids Res.*, vol. 29, pp. 242-245, 2001.

[6] 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.

[7] L. Hirschman, J. C. Park, J. Tsujii, L. Wong, and C. H. Wu, "Accomplishments and challenges in literature data mining for biology," *Bioinformatics*, vol. 18, pp. 1553-1561, 2002.

[8] B. Settles, "ABNER: an open source tool for automatically tagging genes, proteins and other entity names in text," *Bioinformatics*, vol. 21, pp. 3191-3192, 2005.

[9] J. Ding, D. Berleant, D. Nettleton, and E. Wurtele, "Mining MEDLINE: Abstracts, Sentences, or Phrases," *Pacific Symposium on Biocomputing*, vol. 7, pp. 326-337, 2002.

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

[11] R. Apweiler, A. Bairoch, C. H. Wu, W. C. Barker, B. Boeckmann, S. Ferro, E. Gasteiger, H. Huang, R. Lopez, M. Magrane, M. J. Martin, D. A. Natale, C. O'Donovan, N. Redaschi, and L.-S. L. Yeh, "UniProt: the Universal Protein knowledgebase," *Nucl. Acids Res.*, vol. 32, pp. D115-119, 2004.

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

[13] 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.

[14] M. P. Marcus, M. A. Marcinkiewicz, and B. Santorini, "Building a large annotated corpus of English: the penn treebank," *Computational Linguistics*, vol. 19, pp. 313-330, 1993.

[15] 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.

[16] J. Ding, D. Berleant, J. Xu, and A. W. Fulmer, "Extracting Biochemical Interactions from MEDLINE Using a Link Grammar Parser," *Proceedings of the 15th IEEE International Conference on Tools with Artificial Intelligence*, 2003.

[17] Pyysalo, Ginter, Pahikkala, Boberg, Järvinen, and Salakoski, "Evaluation of two dependency parsers on biomedical corpus targeted at protein-protein interactions," *International Journal of Medical Informatics*, vol. 75, pp. 430-442, 2006.

[18] I. Xenarios, L. Salwinski, X. J. Duan, P. Higney, S. M. Kim, and D. Eisenberg, "DIP, the Database of Interacting Proteins: a research tool for studying cellular networks of protein interactions," *Nucleic Acids Res*, vol. 30, pp. 303-5, 2002.

[19] S. Pyysalo, T. Salakoski, S. Aubin, and A. Nazarenko, "Lexical Adaptation of Link Grammar to the Biomedical Sublanguage: a Comparative Evaluation of Three Approaches," *Proceedings of the Second International Symposium on Semantic Mining in Biomedicine*, pp. 60-67, 2006.

[20] E. Tsivtsivadze, T. Pahikkala, J. Boberg, and T. Salakoski, "Locality-Convolution Kernel and Its Application to Dependency Parse Ranking," in *Advances in Applied Artificial Intelligence, 19th International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems, IEA/AIE.*, vol. 4031, *Lecture Notes in Computer Science*, M. A. a. R. Dapoigny, Ed. Annecy, France: Springer, 2006, pp. 610–618.

[21] E. Tsivtsivadze, T. Pahikkala, S. Pyysalo, J. Boberg, A. Myllari, and T. Salakoski, "Regularized Least-Squares for Parse Ranking," in *In proceedings of the 6th International Symposium on Intelligent Data Analysis*, vol. 3646, *Lecture Notes in Computer Science*, A. F. F. e. al., Ed., 2005, pp. 464–474.

[22] 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.

[23] "Obstacles of nomenclature," *Nature*, vol. 389, pp. 1, 1997.

[24] A. Yeh, A. Morgan, M. Colosimo, and L. Hirschman, "BioCreAtIvE Task 1A: gene mention finding evaluation," *BMC Bioinformatics*, vol. 6, pp. S2, 2005.

[25] R. T.-H. Tsai, S.-H. Wu, W.-C. Chou, Y.-C. Lin, D. He, J. Hsiang, T.-Y. Sung, and W.-L. Hsu, "Various criteria in the evaluation of biomedical named entity recognition," *BMC Bioinformatics*, vol. 7, pp. 92, 2006.

[26] K. Seki and J. Mostafa, "A Probabilistic Model for Identifying Protein Names and their Name Boundaries," *Proceedings of the IEEE Computer Society Conference on Bioinformatics* pp. 251, 2003.

[27] K. Fukuda, A. Tamura, T. Tsunoda, and T. Takagi, "Toward information extraction: identifying protein names from biological papers," *Pac Symp Biocomput*, vol. 707, pp. 18, 1998.

[28] F. Olsson, G. Eriksson, K. Franzén, L. Asker, and P. Lidén, "Notions of correctness when evaluating protein name taggers," *Proceedings of the 19th international conference on Computational linguistics-Volume 1*, pp. 1-7, 2002.

[29] L. V. Subramaniam, S. Mukherjea, P. Kankar, B. Srivastava, V. S. Batra, P. V. Kamesam, and R. Kothari, "Information extraction from biomedical literature: methodology, evaluation and an application," *Proceedings of the twelfth international conference on Information and knowledge management*, pp. 410-417, 2003.

[30] C. Nedellec, "Learning Language in Logic-Genic Interaction Extraction Challenge," *Proceedings of The 22nd International Conference on Machine Learning, Bonn, Germany*, 2005.

[31] L. Hirschman, A. Yeh, C. Blaschke, and A. Valencia, "Overview of BioCreAtIvE: critical assessment of information extraction for biology," *BMC Bioinformatics*, vol. 6, pp. S1, 2005.

[32] A. S. Yeh, L. Hirschman, and A. A. Morgan, "Evaluation of text data mining for database curation: lessons learned from the KDD Challenge Cup," *Bioinformatics*, vol. 19, pp. i331-339, 2003.

[33] K. I. M. Jin-Dong, O. Tomoko, Y. T. Yoshimasa Tsuruoka, and N. Collier, "Introduction to the Bio-Entity Recognition Task at JNLPBA," *Proceedings of the Joint Workshop on Natural Language Processing in Biomedicine and its Applications (JNLPBA-2004)*, 2004.

[34] 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.

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

[36] 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.

[37] 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.

[38] 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.

[39] T. Ono, H. Hishigaki, A. Tanigami, and T. Takagi, "Automated extraction of information on protein-protein interactions from the biological literature," *Bioinformatics*, vol. 17, pp. 155-161, 2001.

[40] C. Blaschke and A. Valencia, "The frame-based module of the SUISEKI information extraction system," *IEEE Intelligent Systems*, vol. 17, pp. 14-20, 2002.
[41] T. C. Rindflesch, L. Tanabe, J. N. Weinstein, and L. Hunter, "EDGAR: extraction of drugs, genes and relations from the biomedical literature," *Pac Symp Biocomput*, vol. 2000, pp. 515-524, 2000.
[42] T. Sekimizu, H. S. Park, and J. Tsujii, "Identifying the Interaction between Genes and Gene Products Based on Frequently Seen Verbs in Medline Abstracts," *Genome Inform Ser Workshop Genome Inform*, vol. 9, pp. 62-71, 1998.
[43] 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.
[44] K. Fundel, R. Kuffner, and R. Zimmer, "RelEx - Relation extraction using dependency parse trees," *Bioinformatics*, vol. 23, pp. 365-371, 2007.
[45] S. Novichkova, S. Egorov, and N. Daraselia, "MedScan, a natural language processing engine for MEDLINE abstracts," *Bioinformatics*, vol. 19, pp. 1699-1706, 2003.
[46] 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.
[47] A. Yakushiji, Y. Tateisi, Y. Miyao, and J. Tsujii, "Event extraction from biomedical papers using a full parser," *Pac. Symp. Biocomput*, vol. 6, pp. 408–419, 2001.
[48] 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.
[49] F. Rinaldi, G. Schneider, K. Kaljurand, J. Dowdall, C. Andronis, A. Persidis, and O. Konstanti, "Mining relations in the GENIA corpus," *Proceedings of the Second European Workshop on Data Mining and Text Mining for Bioinformatics*, pp. 61-68, 2004.
[50] 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.
[51] 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.
[52] J. Saric, L. J. Jensen, R. Ouzounova, I. Rojas, and P. Bork, "Extraction of regulatory gene/protein networks from Medline," *Bioinformatics*, pp. bti597, 2005.
[53] 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.
[54] O. Christ, "A Modular and Flexible Architecture for an Integrated Corpus Query System," *Proceedings of COMPLEX'94: 3rd Conference on Computational Lexicography and Text Research*, 1994.

[55] K. B. Cohen, L. Tanabe, S. Kinoshita, and L. Hunter, "A resource for constructing customized test suites for molecular biology entity identification systems," *Linking biological literature, ontologies and databases: tools for users*, pp. 1–8, 2004.

[56] L. Hirschman, "The Evolution of evaluation: Lessons from the Message Understanding Conferences," *Computer Speech & Language*, vol. 12, pp. 281-305, 1998.