AAAI Technical Report FS-12-05
Information Retrieval and Knowledge Discovery in Biomedical Text

# Subgraph Matching-Based Literature Mining for Biomedical Relations and Events

**Haibin Liu**[1] **Vlado Kešelj**[2] **Christian Blouin**[2] **Karin Verspoor**[3]
[1]Colorado Computational Pharmacology, University of Colorado School of Medicine, Aurora, CO, 80045 USA
[2]Faculty of Computer Science, Dalhousie University, Halifax, NS, B3H 4R2 Canada
[3]National ICT Australia, Victoria Research Lab, Melbourne 3010, Australia

## Abstract

Extracting important relations between biological components and semantic events involving genes or proteins from literature has become a focus for the biomedical text mining community. In this paper, we review a subgraph matching-based approach proposed in our previous work for mining relations and events in the biomedical literature. Our subgraph matching algorithm is formally presented, along with a detailed analysis of its complexity. We present three different relation/event extraction tasks in which our approach has been successfully applied. Our approach is of considerable value in extracting highly precise, binary relations when appropriate training data is available.

## Introduction

Recent research in information extraction from the biomedical literature has addressed automatically extracting important relations between biological components such as protein-protein interactions and protein-disease associations, and semantic events involving genes or proteins including gene expression, binding, or regulation events (Kim et al. 2009; 2011). While a relation typically involves a pair of entities with participating roles, linked by a semantic relation type, an event captures the association of multiple participants of varying numbers and with diverse semantic roles (Ananiadou et al. 2010). We will refer to events in this paper.

Automatic extraction of such relations or events has a broad range of biological applications, ranging from support for the annotation of molecular pathways to the automatic enrichment of biological process databases. Since often relations and events can serve as participants in other events, the extraction of such nested event structures also facilitates the construction of complex conceptual networks.

Graphs provide a flexible structure to represent a network and naturally describe the interactions between its components. Therefore, they are a powerful primitive for modeling relations and events. More recently, dependency graphs from syntactic parsing, with their ability to capture long-range dependencies, have shown an advantage in biological relation extraction (Miyao et al. 2009). There are two primary approaches used to integrate dependency graphs with supervised machine learning methods for event extraction: feature-based and kernel-based.

The feature-based approach encodes node tokens and edge dependency types of variable depths of a dependency graph as features to feed learning algorithms, and has been extensively applied for event extraction (Björne et al. 2009). However, these features do not fully capture the rich, structured information of a graph. The kernel-based approach used in conjunction with Support Vector Machines (SVM) is able to use that structure directly. The approach employs a graph kernel, which directly calculates the similarity between two dependency graphs. Various graph kernels have been proposed that compare two graphs according to different characteristics. The shortest path kernel focuses on the shared information on the shortest dependency path between the constituent entities of a relation (Bunescu and Mooney 2005), while the all-paths graph kernel considers weighted shared dependency paths of all possible lengths between words (Airola et al. 2008). They have been applied to extracting protein-protein and drug-drug interactions (Tikk et al. 2010; Thomas et al. 2011a).

Graph matching-based techniques that directly operate on dependency graphs have also proven effective for information extraction in the general English domain. A dependency graph matching module was introduced to compute the text relatedness between student answers and correct answers in assisting the automatic grading of student answers (Mohler, Bunescu, and Mihalcea 2011). Given dependency graphs of question and answer sentences, a method was also proposed to learn graph-based question answering rules by extracting the maximum common subgraph of two graphs, which determines the common information between a question and a sentence containing an answer (Mollá 2006). These approaches achieved accuracy figures competitive with state-of-the-art supervised methods.

In our previous work, we proposed a subgraph matching-based approach to extract events from the biomedical literature (Liu, Blouin, and Keselj 2010). In this paper, we review this approach and demonstrate its generalizability by presenting three relation/event extraction tasks in which our approach has been successfully applied. In the end, we summarize the advantages of this approach. To the best of our knowledge, it is the first graph matching-based approach for extracting relations or events in the biomedical domain.

## Graph-based Event Extraction Method

Interactions among biological entities are characterized in various contexts in the biomedical literature. The same biological processes are often expressed via diverse surface forms in text (Ananiadou et al. 2010).

The underlying assumption of our event extraction approach is that the contextual dependencies of each stated biological relation or event capture a typical context where events of such type are frequently occurring in the biomedical literature. Our approach falls into the category of instance-based reasoning (Alpaydin 2004). Specifically, the key contextual structures are learned from each labeled positive instance and maintained as event rules in the form of subgraphs. When compared against unseen text, rules are generalized according to different matching criteria to identify instances in accordance with rules.

Figure 1 illustrates the overall architecture of our subgraph matching-based event extraction approach with three core components highlighted. In line with most systems (Björne et al. 2009; Airola et al. 2008), our approach focuses on extracting relations or events that are expressed within the boundaries of a single sentence, and those that require information across sentences or articles are not considered. In addition, it is assumed that biological entities involved in the target event have been manually annotated or automatically recognized by upstream procedures.

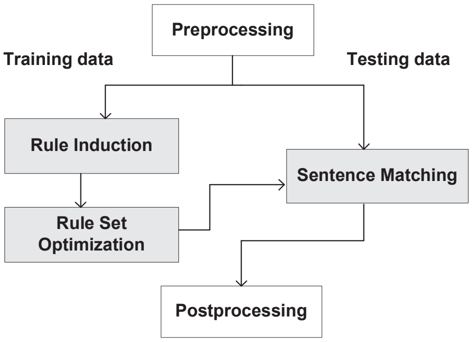


Figure 1: General Architecture of Event Extraction

Several standard preprocessing steps are first completed on both training and testing data. These include sentence segmentation and tokenization, Part-of-Speech tagging, and syntactic parsing that produces dependency graphs for sentences (Klein and Manning 2003).

**Rule Induction** Event rules are learned automatically using the following induction method. Starting with the dependency graph of each training sentence, the shortest dependency path in the undirected version of the graph between certain participants of each annotated event is selected; information regarding their relationship is particularly likely to be carried on this path (Bunescu and Mooney 2005). The union of all such shortest paths is then computed. While the dependencies of the path union is used as the graph representation of the event, a detailed description records the participants of the event, their semantic role labels and the associated nodes in the graph. All the participating biological entities are replaced with a single tag, e.g. "BIO_Entity". As a result, each annotated event is generalized and transformed into a generic graph-based rule. The rule induction algorithm is elaborated in more detail in (Liu, Blouin, and Keselj 2010).

**Sentence Matching** Event extraction from test sentences is achieved by matching the obtained rules to each testing sentence. Since rules and sentences all possess a graph representation, event recognition becomes a subgraph matching problem, to identify a subgraph isomorphic to a rule graph within the graph of a testing sentence. The subgraph matching problem in our work is defined as follows.

**Definition 1.** A rule graph $G_r = (V_r, E_r)$ is isomorphic to a subgraph of a sentence graph $G_s = (V_s, E_s)$, denoted by $G_r \cong S_s \subseteq G_s$, if there is an injective mapping $f : V_r \rightarrow V_s$ such that, for every directed pair of nodes $v_i, v_j \in V_r$, if $(v_i, v_j) \in E_r$ then $(f(v_i), f(v_j)) \in E_s$, and the edge label of $(v_i, v_j)$ is the same as the edge label of $(f(v_i), f(v_j))$.

We designed a subgraph matching algorithm to perform this sentence matching task (Liu, Blouin, and Keselj 2010). We present the formal algorithm and give a detailed analysis of its complexity in the next section.

**Rule Set Optimization** Typical of instance-based reasoners, the accuracy of rules with which to compare an unseen sentence is crucial to the success of our approach. Although rules are induced from positively labeled events, when the graph representation of a rule is detected in previously unseen text, the encoded contextual dependencies may not always contain a valid event. For instance, a *Gene_expression* rule encoding a dependency relation for "Tax expression" should not produce a *Gene_expression* event for the phrase "Tax expression vector" even though they share a same dependency, because "Tax expression" is used as an adjective to describe "vector" in this context. Such matches result in false positive events.

Therefore, we measured the accuracy of each rule $r_i$ in terms of its prediction result via Eq.(1). Each rule is compared against training sentences using the subgraph matching approach, leaving out the sentence from which the rule was learned. For rules that produce at least one prediction, we ranked them by $ACC(r_i)$ and excluded the ones with a $ACC(r_i)$ ratio lower than an empirical threshold, e.g. 1:4. We assume that these rules will produce false positive predictions on unseen text if they are retained in the rule set. Rules that do not make predictions are kept as they may potentially contribute to the testing data.

$$ACC(r_i) = \frac{\#correct\_predictions\_by\_r_i}{\#total\_predictions\_by\_r_i} \qquad (1)$$

This performance-based evaluation leads to an optimized rule set. It is incorporated into our event extraction framework as this component substantially improves the precision of the method (Liu, Komandur, and Verspoor 2011).

Finally, post-processing is performed to transform raw sentence matching results into the required format according to the event extraction task.

## Subgraph Matching Algorithm[1]

The subgraph matching problem is NP-complete (Garey and Johnson 1979). Since on average there are about 24 words in a sentence in the biomedical text (Kim et al. 2003), the dependency graphs of rules and sentences involved in our matching process are small. Therefore, we designed a simple subgraph matching algorithm using a backtracking approach (Liu, Blouin, and Keselj 2010). The main and the recursive part of the algorithm are formalized in Algorithm 1 and 2.

**Algorithm 1** Main algorithm

```
Input: Dependency graph of a testing sentence $s$, $G_s = (V_s, E_s)$
    where $V_s$ is the set of nodes and $E_s$ is the set of edges of the
    graph; a finite set of rules $R = \{r_1, \cdots, r_i, \cdots\}$, where $r_i =$
    $(e_i,\ G_{r_i})$. $G_{r_i} = (V_{r_i},\ E_{r_i})$ is the dependency graph of $r_i$.
Output: $MR$ : a set of rules from $R$ matched with $s$ together with
    the injective mapping
Main algorithm:
 1: $MR \leftarrow \emptyset$
 2: for all $r_i \in R$ do
 3:     $st_{r_i} \leftarrow$ StartNode$(G_{r_i})$          //StartNode finds the start
 4:     //node $st_{r_i}$ of the rule graph $G_{r_i}$
 5:     $ST_s \leftarrow \{st_{s_1}, st_{s_2}, \cdots, st_{s_j}, \cdots\}$
 6:     //$ST_s$ : the set of start nodes of the sentence graph $G_s$
 7:     for all $st_{s_j} \in ST_s$ do
 8:         create an empty stack $\sigma$ and push $(st_{r_i}, st_{s_j})$ onto
 9:         the stack $\sigma$
10:         $IM \leftarrow \emptyset$          //$IM$ : records of injective matches
11:         //between nodes in $G_{r_i}$ and $G_s$
12:         call MatchNode$(\sigma, rIM, G_{r_i}, G_s)$
13:         //$rIM$ : reference of $IM$
14:         if MatchNode returned TRUE then
15:             $MR \leftarrow MR \cup \{r_i$ with $IM\}$
16: return $MR$
```

**Algorithm 2** Recursive subroutine

```
Recursive subroutine: MatchNode$(\sigma, rIM_{parent}, G_{r_i}, G_s)$
 1: $IM_{current} \leftarrow IM_{parent}$          //assign $IM_{parent}$ from the
 2: //parent level to the current $IM_{current}$
 3: while stack $\sigma$ is not empty do
 4:     pop node pair $(v_r, v_s)$ from stack $\sigma$
 5:     if an injective match between $v_r$ and $v_s$ already exists
 6:     in $IM_{current}$ then
 7:         do nothing
 8:     else if an injective match is possible between $v_r$ and
 9:     $v_s$ then
10:         $IM_{current} \leftarrow IM_{current} \cup \{$ the match between
11:         $v_r$ and $v_s$ $\}$
12:     else
13:         return FALSE
14:     for all edges $e_r$ adjacent to node $v_r$ in $G_{r_i}$ do
15:         let $(v_r, n_r)$ be the edge $e_r$
16:         for all edges $e_s$ adjacent to node $v_s$ in $G_s$ do
17:             let $(v_s, n_s)$ be the edge $e_s$
18:             if $e_r$ and $e_s$ share same direction and label then
19:                 $S \leftarrow S \cup n_s$          //$S$ : the set of candidate
20:                 //nodes for matching $n_r$
21:         for all $n_s \in S$ do
22:             if an injective match between $n_r$ and $n_s$
23:             already exists in $IM_{current}$ then
24:                 go to Line 14 and proceed with next edge $e_r$
25:             else if an injective match is possible between
26:             $n_r$ and $n_s$ then
27:                 $\sigma_n \leftarrow \sigma$          //copy $\sigma$ to a new stack $\sigma_n$
28:                 push $(v_r, v_s, n_r, n_s)$ onto the stack $\sigma_n$
29:                 call MatchNode$(\sigma_n, rIM_{current}, G_{r_i}, G_s)$
30:                 //$rIM_{current}$ : reference of $IM_{current}$
31:                 if MatchNode returned TRUE then
32:                     $IM_{parent} \leftarrow IM_{current}$
33:                     //update $IM_{parent}$ using $IM_{current}$
34:                     return TRUE
35:         return FALSE
36: $IM_{parent} \leftarrow IM_{current}$
37: return TRUE
```

The backtracking ability of the algorithm allows the matching process to recover from initial incorrect matches and continue to proceed until the correct subgraph is identified. An example of the backtracking process when matching a rule graph with a sentence graph is illustrated in Figure 2. The matches are highlighted by dotted lines.

The complexity of Algorithm 1 is exponential, as we could expect since the problem of subgraph matching is known to be NP-hard. However, we have observed that the algorithm is relatively efficient in practice and we have successfully run it on several event extraction tasks. We show that this efficient performance in practice can be expected. Let us assume that the sentence graph $G_s$ and the rule graph $G_{r_i}$ have a total of $n$ vertices and $m$ edges, and the vertex degree (number of adjacent edges) is always less than or equal $k$. The main algorithm has two nested loops so it calls the recursive part MatchNode $O(|R| \cdot n)$ times. When calling MatchNode, the main source of inefficiency is the occurrence of several edges with the same label, adjacent to one node. This is more an exception in realistic dependency parses than the rule. If we had two graphs with no adjacent same-label edges, MatchNode would be called for each pair of matchable nodes, which makes $O(n)$ invocations. The nested loops iterate $O(nk^2)$ times. Since line 27 requires $O(n)$ time to copy the stack, the total time would be $O(|R| \cdot n^3k^2)$ time. However, if same-label adjacent edges are present, the algorithm may backtrack to try to match each edge with $k$ possible edges in the other graph, which gives $O(k^n)$ possible invocations of MatchNode, with the total worst-case algorithm complexity $O(|R| \cdot n^2k^n)$. In practice, it only takes the algorithm less than a second to return the results for each sentence.

When matching between graphs, various matching features can be considered, resulting in different matching criteria. The features include edge features (E) which are edge labels and edge directions, and node features which are POS tags (P) and all tokens (A), ranging from the least specific matching criterion, E, to the much stricter criterion, A. For each sentence, the algorithm returns all the matched rules together with the injective mappings from rule nodes to sentence tokens. Events are then extracted by applying the descriptions of tokens in each matched rule (e.g. role labels) to the corresponding tokens of the sentence.

[1] The Java implementation of our algorithm is released as open source software that can be downloaded via http://esmalgorithm.sourceforge.net/.

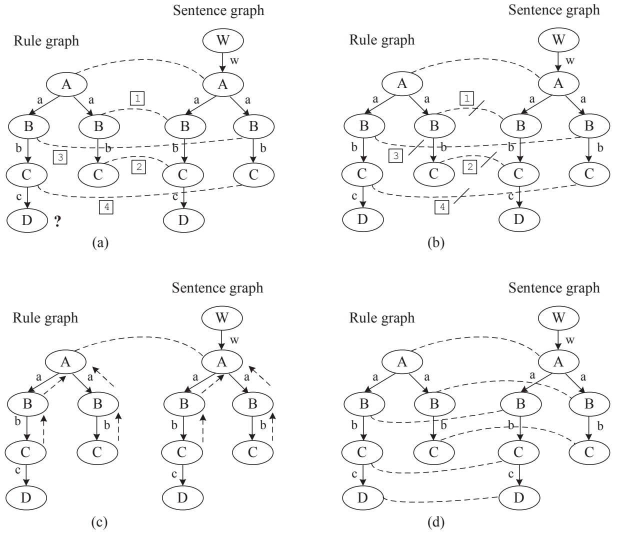


Figure 2: Example of Backtracking Process (a) initial injective matches (b) wrong matches detected (c) backtracking to the A node (d) correct matches found

The subgraph matching algorithm can also be used to determine isomorphism relationships between rules by examining whether the graph representations of rules are subgraph isomorphic to each other according to a matching criterion. Although duplicate events produced by isomorphic rules will be removed eventually via post-processing, keeping only graphically unique rules can significantly reduce the size of the rule set to be matched, thus improving the overall efficiency of the event extraction.

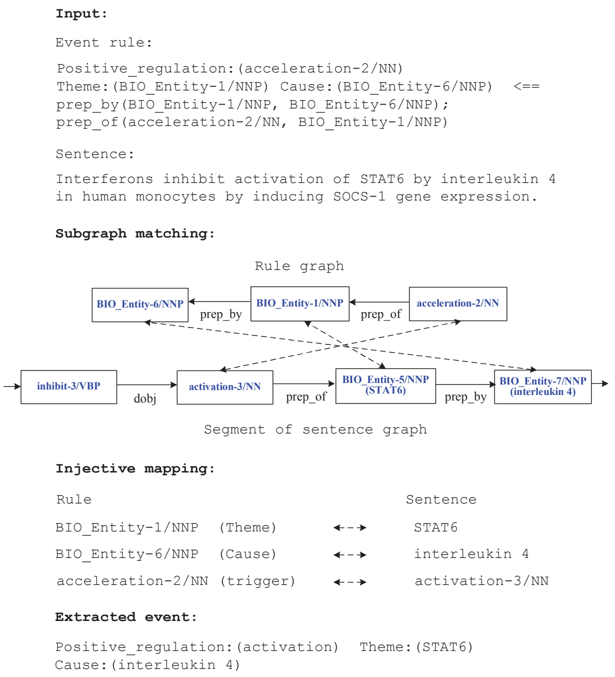
## Application of Graph-based Event Extraction

In this section, we demonstrate three successful biomedical applications of our event extraction approach: BioNLP shared tasks, Protein-Residue association detection and Protein-Protein interaction identification.

**BioNLP Shared Tasks** The two BioNLP shared tasks focused on the recognition of biological events particularly on proteins in the literature with the gold protein annotation given (Kim et al. 2009; 2011). When a biological event is described in text, it can be recognized by the event type, the event trigger, and one or more event arguments.

For each gold event, the shortest dependency path connecting the event trigger to each event argument is extracted when learning event rules from training sentences. For complex events such as regulation events that take a sub-event as an argument, the shortest path is extracted so as to connect the trigger of the main event to the trigger of the sub-event. The resulting rules are categorized into different event types.

Figure 3 presents a simple example of the event extraction process by matching an event rule to a sentence to extract a *Positive_regulation* event in the sentence. The matching criteria in the example, "E+P", require that edges be matched if they share the same direction and edge label while nodes be matched as long as the POS tags of the tokens are identical.



```
Input:

Event rule:

Positive_regulation:(acceleration-2/NN)
Theme:(BIO_Entity-1/NNP) Cause:(BIO_Entity-6/NNP)  <==
prep_by(BIO_Entity-1/NNP, BIO_Entity-6/NNP);
prep_of(acceleration-2/NN, BIO_Entity-1/NNP)

Sentence:

Interferons inhibit activation of STAT6 by interleukin 4
in human monocytes by inducing SOCS-1 gene expression.

Subgraph matching:
```

```
Injective mapping:

Rule                                           Sentence

BIO_Entity-1/NNP  (Theme)         <-->         STAT6

BIO_Entity-6/NNP  (Cause)         <-->         interleukin 4

acceleration-2/NN (trigger)       <-->         activation-3/NN

Extracted event:

Positive_regulation:(activation)  Theme:(STAT6)
Cause:(interleukin 4)
```

Figure 3: Biological Event Extraction Process

We applied our event extraction approach to both shared tasks. Table 1 reports the best performance by our approach on the testing set of the GENIA Event (GE) task of BioNLP-ST 2011, which subsumes the testing data of BioNLP-ST 2009. It is evaluated by the primary metric of the tasks via the official online evaluation.

| Event type | Rec.(%) | Prec.(%) | F(%) |
|---|---|---|---|
| Gene_expression (1002) | 62.08 | 89.24 | 73.22 |
| Transcription (174) | 39.08 | 81.93 | 52.92 |
| Protein_catabolism (15) | 53.33 | 100.00 | 69.57 |
| Phosphorylation (185) | 61.08 | 88.28 | 72.20 |
| Localization (191) | 31.94 | 95.31 | 47.84 |
| [SVT-TOTAL] (1567) | 55.65 | 88.98 | 68.47 |
| Binding (491) | 25.66 | 61.46 | 36.21 |
| [EVT-TOTAL] (2058) | 48.49 | 84.22 | 61.55 |
| Regulation (385) | 20.00 | 42.54 | 27.21 |
| Positive_regulation (1443) | 31.32 | 57.00 | 40.43 |
| Negative_regulation (571) | 24.87 | 40.11 | 30.70 |
| [REG-TOTAL] (2399) | 27.97 | 50.53 | 36.01 |
| [ALL-TOTAL] (4457) | 37.45 | 66.41 | 47.89 |

Table 1: GE results of "E+P*+A*" on testing set by "Approximate Span /Approximate Recursive Matching"

The graph matching criteria "E+P*+A*" requires that the edge features (E), the relaxed POS tags (P*) and the lemmatized forms of all tokens (A*) be exactly the same. The relaxed POS allows the plural form of nouns to match with the singular form, and the conjugations of verbs to match with each other. Lemmatization is performed by the BioLemmatizer (Liu et al. 2012) on every pair of node tokens to be matched to allow tokens that share a same lemma to match.

Our subgraph matching-based event extraction method clearly shows an overall superior precision over all the participating teams of BioNLP-ST 2011, of which only three individual systems achieved a precision in the 60% range. Particularly, the precision of five simple events that only involve a trigger and a theme is approaching 90%, nearly 9% higher than that of the best performing system. This indicates that event rules automatically learned and optimized over training data generalize well to the unseen text. Whenever the graph representation of a rule is detected in testing data, the rule has the ability to identify precisely a corresponding event. Considering that the precision outperforms the system relying on manually developed patterns (Kilicoglu and Bergler 2011), it indicates that learned rules can be even more accurate than human-coded rules.

However, the overall performance is limited by the lower coverage. While 55.7% of simple events are captured, the recall of more complex events such as *Binding* and regulation events are much lower. This suggests that the lexical information and syntactic dependencies expressed in rules are not sufficient enough to cover more complex event contexts where multiple participants are involved. These events often require a long dependency path from trigger to arguments.

One way to improve recall is to enrich the rule set with rules learned via *distant supervision* to help cover diverse event contexts. Distant supervision automatically creates training instances by heuristically matching the existing knowledge to corresponding text (Craven and Kumlien 1999). Next, we present two applications that integrate distant supervision into our relation extraction framework.

**Protein-Residue Association** In three-dimensional protein structures, the appearance of certain amino acid residues at key structural positions plays a central role in protein function, for instance enabling ligand or substrate binding. For proteins of therapeutic importance, identifying these protein residues as potential targets is a key early step in drug design. Text mining has been shown to play an important role in such protein function prediction (Verspoor et al. 2012). Our event extraction approach was applied to extract protein-residue associations embedded in the biomedical literature (Ravikumar et al. 2012).

Instead of manually curated annotations, sentences that contain high confidence protein-residue relationships were prepared via distant supervision using Protein Data Bank (PDB) as the biological knowledge source. Sentences in which at least a protein and an amino acid co-occur were selected from abstracts of the primary references for the PDB entries. These sentences are further filtered to retain only those that contain physically validated relationships, i.e., the protein-residue co-occurrence can be substantiated by a physical match of the particular residue to the mentioned protein according to its PDB record. For the sentence “CTP binding affects the conformation of Arg80, and the Arg80 conformation in the UPRTase-UMP-CTP complex leaves no room for binding of the substrate PRPP.”, the protein-residue pair (UPRTase-Arg80) is validated via the PDB entry “1xtv”, with PMID-15654744 as the primary citation. Association rules are then induced from these sentences by extracting the shortest paths connecting association arguments.

Our approach achieved a 80% F-score in extracting protein-residue associations (Ravikumar et al. 2012) from the Nagel corpus, in which proteins and amino acid residues are pre-annotated (Nagel et al. 2009), with a 72% recall and a 90% precision, surpassing previously published methods. Distant supervision helps to relax the reliance of rule induction on the curated annotation. Taking advantage of a much broader set of training instances, more rules are reliably learned to cover diverse relation contexts, thus improving the coverage of our approach.

**Protein-Protein Interaction** Protein-protein interactions (PPI) form the basis for a vast majority of cellular processes, including signal transduction and transcriptional regulation. The study of these interactions is fundamental to the understanding of biological systems. Literature-based PPI identification has been an active research area for the biomedical text mining community.

Our method was successfully adopted to serve as the basis for extracting protein-protein interactions (Thomas et al. 2011b). Distant supervision is performed to create training sentences for the generation of rules. A database of PPIs, IntAct, is used against all sentences in Medline and PMC with proteins automatically tagged and normalized to select those sentences containing any of the protein-protein pairs. Instead of ranking rules, a set of rule generalizers and filters is proposed to systematically optimize the rule set.

When evaluated on five benchmark PPI corpora (AIMed, BioInfer, HPRD50, IEPA, and LLL), our approach achieves a comparable performance to state-of-the-art machine learning-based PPI extraction methods. In particular, it obtains the second best F-score among all evaluated approaches on the largest PPI corpus BioInfer. This confirms the effectiveness of distant supervision in our approach.

## Conclusion

In this paper, we have reviewed a subgraph matching-based event extraction approach, and demonstrated its generalizability via three successful applications of event extraction in the biomedical domain. This approach has a number of advantageous features.

First, characterized by a high precision, our approach is a preferable choice when accurate information about biological processes is emphasized. It works particularly well on extracting binary relations (including events containing only two participants) with training data where biological entities of the target relation are pre-annotated. Second, the coverage of the approach can be effectively increased by integrating distant supervision. Meanwhile, rules learned from co-mentions of pairs of entities known to interact are not prone to over-fitting to an annotated training corpus, thus more generalizable across different datasets (Thomas et al. 2011b). In contrast, most state-of-the-art machine learning methods for relation extraction show large performance differences depending on whether or not the evaluation and training instances are taken from the same corpus (Tikk et al. 2010). Third, our approach is easily adapted to different event extraction tasks. Its generalizability has been demonstrated via three biomedical applications with various requirements and diverse contexts. The task-specific adaptation only involves specifying the type of the targeted relation, e.g. protein-residue association, and is therefore trivial. Moreover, existing ontological resources can be naturally applied to the matching process between graph nodes to improve the overall event extraction performance. Fourth, analyzing extraction errors of the approach is more straightforward compared to SVM-based supervised learning methods as a wrong match can be pinpointed to the specific rule producing it and then corrected.

## References

Airola, A.; Pyysalo, S.; Björne, J.; Pahikkala, T.; Ginter, F.; and Salakoski1, T. 2008. All-paths graph kernel for protein-protein interaction extraction with evaluation of cross-corpus learning. *BMC Bioinformatics* 9 Suppl 11:s2.

Alpaydin, E. 2004. *Introduction to Machine Learning*. MIT Press.

Ananiadou, S.; Pyysalo, S.; Tsujii, J.; and Kell, D. B. 2010. Event extraction for systems biology by text mining the literature. *Trends in Biotechnology* 28(7):381–390.

Björne, J.; Heimonen, J.; Ginter, F.; Airola, A.; Pahikkala, T.; and Salakoski, T. 2009. Extracting complex biological events with rich graph-based feature sets. In *BioNLP '09: Proceedings of the Workshop on BioNLP*, 10–18. Association for Computational Linguistics.

Bunescu, R. C., and Mooney, R. J. 2005. A shortest path dependency kernel for relation extraction. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, 724–731.

Craven, M., and Kumlien, J. 1999. Constructing biological knowledge bases by extracting information from text sources. In *Proceedings of the Seventh International Conference on Intelligent Systems for Molecular Biology*, 77–86. AAAI Press.

Garey, M. R., and Johnson, D. S. 1979. *Computers and Intractability; A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co.

Kilicoglu, H., and Bergler, S. 2011. Adapting a general semantic interpretation approach to biological event extraction. In *Proceedings of the BioNLP Shared Task 2011 Workshop*, BioNLP Shared Task '11, 173–182. Association for Computational Linguistics.

Kim, J.-D.; Ohta, T.; Teteisi, Y.; and Tsujii, J. 2003. Genia corpus - a semantically annotated corpus for bio-textmining. *Bioinformatics* 19(suppl. 1):i180–i182.

Kim, J.-D.; Ohta, T.; Pyysalo, S.; Kano, Y.; and Tsujii, J. 2009. Overview of BioNLP'09 shared task on event extraction. In *Proceedings of BioNLP Shared Task 2009 Workshop*, 1–9. Association for Computational Linguistics.

Kim, J.-D.; Pyysalo, S.; Ohta, T.; Bossy, R.; Nguyen, N.; and Tsujii, J. 2011. Overview of BioNLP shared task 2011. In *Proceedings of BioNLP Shared Task 2011 Workshop*, 1–6. Association for Computational Linguistics.

Klein, D., and Manning, C. D. 2003. Accurate unlexicalized parsing. In *ACL '03: Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*, 423–430. Association for Computational Linguistics.

Liu, H.; Christiansen, T.; Baumgartner, W. A.; and Verspoor, K. 2012. Biolemmatizer: a lemmatization tool for morphological processing of biomedical text. *Journal of Biomedical Semantics* 3(3).

Liu, H.; Blouin, C.; and Keselj, V. 2010. Biological event extraction using subgraph matching. In *Proceedings of the 4th International Symposium on Semantic Mining in Biomedicine (SMBM-2010)*.

Liu, H.; Komandur, R.; and Verspoor, K. 2011. From graphs to events: A subgraph matching approach for information extraction from biomedical text. In *Proceedings of BioNLP Shared Task 2011 Workshop*, 164–172. Association for Computational Linguistics.

Miyao, Y.; Sagae, K.; Saetre, R.; Matsuzaki, T.; and Tsujii, J. 2009. Evaluating contributions of natural language parsers to protein–protein interaction extraction. *Bioinformatics* 25(3):394–400.

Mohler, M.; Bunescu, R.; and Mihalcea, R. 2011. Learning to grade short answer questions using semantic similarity measures and dependency graph alignments. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, 752–762. Association for Computational Linguistics.

Mollá, D. 2006. Learning of graph-based question answering rules. In *Proceedings of the First Workshop on Graph Based Methods for Natural Language Processing*, 37–44. Association for Computational Linguistics.

Nagel, K.; Jimeno-Yepes, A.; Rebholz-Schuhmann, D.; et al. 2009. Annotation of protein residues based on a literature analysis: cross-validation against uniprotkb. *BMC Bioinformatics* 10(S-8):4.

Ravikumar, K.; Liu, H.; Cohn, J.; Wall, M. E.; and Verspoor, K. 2012. Literature mining of protein-residue associations with graph rules learned through distant supervision. *Journal of Biomedical Semantics* 3 (Suppl. 3):S2.

Thomas, P.; Neves, M.; Solt, I.; Tikk, D.; and Leser, U. 2011a. Relation extraction for drug-drug interactions using ensemble learning. In *Proceedings of DDIExtraction-2011 challenge task*, 11–18.

Thomas, P.; Pietschmann, S.; Solt, I.; Tikk, D.; and Leser, U. 2011b. Not all links are equal: Exploiting dependency types for the extraction of protein-protein interactions from text. In *Proceedings of BioNLP 2011 Workshop*, 1–9. Association for Computational Linguistics.

Tikk, D.; Thomas, P.; Palaga, P.; Hakenberg, J.; and Leser, U. 2010. A comprehensive benchmark of kernel methods to extract protein–protein interactions from literature. *PLoS Computational Biology* 6:e1000837.

Verspoor, K.; Cohn, J.; Ravikumar, K.; and Wall, M. E. 2012. Text mining improves prediction of protein functional sites. *PLoS ONE* 7(2):e32171.