From Graphs to Events: A Subgraph Matching Approach for Information Eextraction from Biomedical Text

Haibin Liu, Ravikumar Komandur, Karin Verspoor
Center for Computational Pharmacology
University of Colorado School of Medicine
PO Box 6511, MS 8303, Aurora, CO, 80045 USA

Abstract

We participated in the BioNLP Shared Task 2011, addressing the GENIA event extraction (GE) and the Epigenetics and Post-translational Modifications (EPI) tasks. A graph-based approach is employed to automatically learn rules for detecting biological events in the life-science literature. The event rules are learned by identifying the key contextual dependencies from full syntactic parsing of annotated text. Event recognition is performed by searching for an isomorphism between event rules and the dependency graphs of sentences in the input texts. While we explored methods such as performance-based rule ranking to improve precision, we merged rules across multiple event types in order to increase recall.

We achieved a 41.13% F-score in detecting events of nine types in the Task 1 of the GE task, and a 52.67% F-score in identifying events across fifteen types in the core task of the EPI task. Our performance on both tasks is comparable to the state-of-the-art systems. Our approach does not require any external domain-specific resources. The consistent performance on the two tasks supports the claim that the method generalizes well to extract events from different domains where training data is available.

1 Introduction

Recent research in information extraction in the biological domain has focused on extracting semantic events involving genes or proteins, such as binding events or post-translational modifications. To date, most of the biological knowledge about these events has only been available in the form of unstructured text in scientific articles (Abulaish and Dey, 2007; Ananiadou et al., 2010).

When a biological event is described in text, it can be analyzed by recognizing its type, the trigger that signals the event, and one or more event arguments. The BioNLP-ST 2009 (Kim et al., 2009) focused on the recognition of semantically typed, complex events in the biological literature. Although the best-performing system achieved a 51.95% F-score in identifying events across nine types, only 4 of the rest 23 participating teams obtained an F-score in the 40% range. This suggests that the problem of biological event extraction is difficult and far from solved.

Graphs provide a powerful primitive for modeling biological data such as pathways and protein interaction networks (Tian et al., 2007; Yan et al., 2006). More recently, the dependency representations obtained from full syntactic parsing, with its ability to reveal long-range dependencies, has shown an advantage in biological relation extraction over the traditional Penn Treebank-style phrase structure trees (Miyao et al., 2009). Since the dependency representation maps straightforwardly onto a directed graph, operations on graphs can be naturally applied to the problem of biological event extraction.

We participated in the BioNLP-ST 2011 (Kim et al., 2011a), and applied a graph matching-based approach (Liu et al., 2010) to tackling the Task 1 of the GENIA event extraction (GE) task (Kim et al., 2011b), and the core task of the Epigenetics and Post-translational Modifications (EPI) task (Ohta et al., 2011), two main tasks of the BioNLP-ST 2011. Event recognition is performed by searching for an isomorphism between dependency representations of automatically learned event rules and complete sentences in the input texts. This process is treated as a subgraph matching problem, which corresponds to the search for a subgraph isomorphic to a rule graph within a sentence graph. While we explored methods such as performance-based rule ranking to improve the precision of the GE and EPI tasks, we merged rules across multiple event types in order to increase the recall of the EPI task.

The rest of the paper is organized as follows: In Section 2, we introduce the BioNLP Shared Task 2011. Section 3 describes the subgraph matching-based event extraction method. Section 4 and Section 5 elabo-

Proceedings of BioNLP Shared Task 2011 Workshop, pages 164–172,
Portland, Oregon, USA, 24 June, 2011.

rate the implementation details and our performance respectively. Finally, Section 6 summarizes the paper and introduces future work.

2 BioNLP Shared Task 2011

The BioNLP-ST 2011 is the extension of the BioNLP-ST 2009 that focused on the recognition of events in the biological literature. The BioNLP-ST 2011 extends the previous task in three directions: the type of the investigated text, the domain of the subject, and the targeted event types. As a result, the shared task was organized into four independent tasks: GENIA Event Extraction Task (GE), Epigenetics and Post-translational Modifications Task (EPI), Infectious Diseases Task (ID) and Bacteria Track.

The definition of the GE task remained the same as the BioNLP-ST 2009. However, additional annotated texts that come from full papers were provided together with the dataset of the 2009 task to generalize the task from PubMed abstracts to full text articles. The primary task of the GE task was to detect biological events of nine types such as protein binding and regulation, given the annotation of protein names. It was required to extract type, trigger, and primary arguments of each event. This task is an example of extraction of semantically typed, complex events for which the arguments can also be other events. Such embedding results in a nested structure that captures the underlying biological statements more accurately.

Different from the subject domain of the GE task on transcription factors in human blood cells, the EPI task focused on events related to epigenetic change, including DNA methylation and histone modification, as well as other common post-translational protein modifications. The core task followed the definition for Phosphorylation event extraction in the 2009 task, and extended that basic event type to a total of fifteen types including both positive and negative variants, for example *Acetylation* and *Deacetylation*. The task dataset was prepared from relevant PubMed abstracts, with additional evidence sentences from databases such as PubMeth (Ongenaert et al., 2007). Given the annotation of protein names, the core task required to extract type, trigger, and primary arguments of each event.

We focused on the primary task of GE and the core task of EPI, and tackled the event extraction problem in both cases using a graph matching-based method.

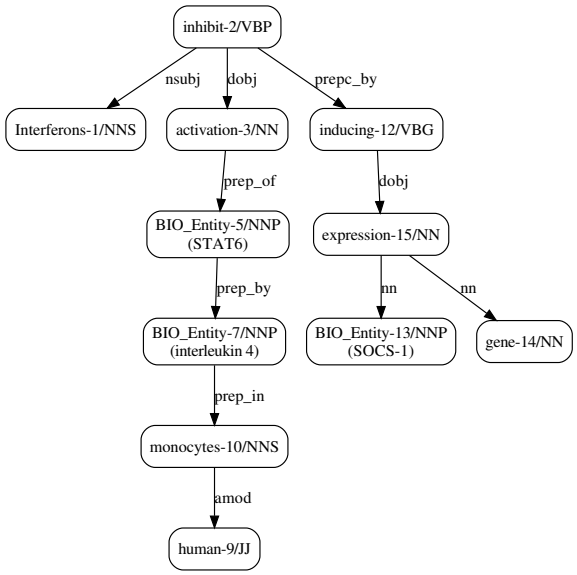

Figure 1: Dependency Graph Example

3 Subgraph Matching-based Event Extraction

3.1 Dependency Representation

The dependency representation of a sentence is formed by tokens in the sentence and binary relations between them. A single dependency relation is represented as *relation*$(\mathit{governor}, \mathit{dependent})$, where *governor* and *dependent* are tokens, and *relation* is a type of the grammatical dependency relation. This representation is essentially a labeled directed graph, which is named *dependency graph* and defined as follows:

Definition 1. A dependency graph is a pair of sets $G = (V, E)$, where V is a set of nodes that correspond to the tokens in a sentence, and E is a set of directed edges, for which the edge labels are types of dependency relations between the tokens, and the edge direction is from *governor* to *dependent* node.

Figure 1 illustrates the dependency graph for the sentence: "Interferons inhibit activation of STAT6 by interleukin 4 in human monocytes by inducing SOCS-1 gene expression." (MEDLINE: 10485906). The token number in the sentence is appended to each token in order to differentiate identical tokens that co-occur in a sentence. All the protein names in the sentence have been replaced with a unified tag "BIO_Entity". The POS tag of each token is noted. "BIO_Entity" tokens are uniformly tagged as proper nouns.

3.2 Event Rule Induction

The premise of our work is that there is a set of frequently occurring event rules that match a majority of

stated events about protein biology. We consider that an event rule encodes the detailed description and characterizes the typical contextual structure of a group of biological events. The rules are learned from labeled training sentences using a graph-based rule induction method (Liu et al., 2010), and we briefly describe the algorithm as follows.

Starting with the dependency graph of each training sentence, edge directions are first removed so that the directed graph is transformed into an undirected graph, where a path must exist between any two nodes since the graph is always connected. For each gold event, the shortest dependency path in the undirected graph connecting the event trigger nodes to each event argument node is selected. The union of all shortest dependency paths is then computed, and the original directed dependency representation of the path union is retrieved and used as the graph representation of the event.

For multi-token event triggers, the shortest dependency path connecting the node of every trigger token to the node of each event argument is selected, and the union of the paths is then computed for each trigger. For regulation events, when a sub-event is used as an argument, only the type and the trigger of the sub-event are preserved as the argument of the main events. The shortest dependency path is extracted so as to connect the trigger nodes of the main event to the trigger nodes of the sub-event. In case that there exists more than one shortest path, all of the paths are considered. As a result, each gold event is transformed into the form of a biological event rule. The algorithm is elaborated in more detail in (Liu et al., 2010). The obtained rules are categorized in terms of the event types of the tasks.

3.3 Sentence Matching

We attempted to match event rules to each testing sentence to extract events from the sentence using a sentence matching approach. Since the event rules and the sentences all possess a dependency graph, the matching process is a subgraph matching problem, which corresponds to the search for a subgraph isomorphic to an event rule graph within the graph of a testing sentence. The subgraph matching problem is also called *subgraph isomorphism*, defined in this work as follows:

Definition 2. An event 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 $(f(v_i), f(v_j))$.

The subgraph isomorphism problem is NP-complete (Cormen et al., 2001). A number of algorithms have been designed to tackle the problem of subgraph isomorphism in different applications (Ullmann, 1976; Cordella et al., 2004; Pelillo et al., 1999). Considering that the graphs of rules and sentences involved in the matching process are small, a simple subgraph matching algorithm using a backtracking approach (Liu et al., 2010) was used in this work. It is named "Injective Graph Embedding Algorithm" and designed based on the Huet's graph unification algorithm (Huet, 1975). The formalized algorithm and the detailed description are given in (Liu et al., 2010).

When matching between graphs, different combinations of matching features can be applied, resulting in different matching criteria. The features include edge features (E) which are edge label and edge direction, and node features which are POS tags (P), trigger tokens (T), 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 corresponding injective mappings from rule nodes to sentence tokens. Biological events are then extracted by applying the event descriptions of tokens in each matched rule consisting of the type, the trigger and the arguments to the corresponding tokens of the sentence.

4 Implementation

4.1 Preprocessing

The same preprocessing steps as in (Liu et al., 2010) are completed on the datasets of the GE and the EPI tasks before performing text mining strategies. These include sentence segmentation and tokenization, Part-of-Speech tagging, and sentence parsing.

The Stanford unlexicalized natural language parser (version 1.6.5), which includes Genia Treebank 1.0 (Ohta et al., 2005) as training material, is used to analyze the syntactic structure of the sentences. The parser returns a dependency graph for each sentence.

4.2 Rule Induction and Sentence Matching

For each gold event, the shortest path in the undirected graph connecting the event trigger to each event argument is extracted using Dijkstra's algorithm (Cormen et al., 2001) with equal weight for edges.

Sentence matching is performed and the raw matching results are then postprocessed based on the specifications of the shared task, such as event trigger cannot

be a protein name or another event.

5 Results and Evaluation

This section presents our results on the GE and the EPI tasks (Kim et al., 2011b; Ohta et al., 2011) respectively. Different experimental methods in processing the obtained event rules are described for the purpose of improving the precision of both tasks and increasing the recall of the EPI task.

5.1 GE task

5.1.1 Preprocessing Results

For training data, only sentences that contain at least one protein and one event are considered candidates for further processing. For testing data, candidate sentences contain at least one protein. Our event recognition method focuses on extracting events from sentences. Therefore, only sentence-based events are considered in this work. Table 1 presents some statistics of the preprocessed datasets.

Attributes Counted	Training	Dev.	Testing
Abstracts&Full articles	908	259	347
Total sentences	8,759	2,954	3,437
Candidate sentences	3,615	1,989	2,353
Total events	10,287	3,243	4,457
Sentence-based events	9,583	3,058	hidden

Table 1: Statistics of GE dataset

We were able to build event rules for 9,414 gold events. Gold events in which the event trigger and an event argument are not connected by a path in the undirected dependency graph of the sentence could not be transformed into a biological event rule. After removing duplicate rules, we obtained 8,677 event rules, which are distributed over nine event types. The rules that are isomorphic to each other in terms of their graph representation are not filtered at this stage as the duplicate events they produce will be removed eventually to prepare the annotations for the shared task.

5.1.2 Probability-based rule refining

We observed that some event rules of an event type overlap with rules of other event types. For instance, a *Transcription* rule is isomorphic to a *Gene_expression* rule in terms of the graph representation and they also share a same event trigger token. In fact, tokens like "gene expression" and "induction" are used as event trigger of both *Transcription* and *Gene_expression* in training data. Therefore, the detection of some *Gene_expression* events is always accompanied by certain *Transcription* events. This will have detrimental effects on the precision of both *Transcription* and *Gene_expression* event types.

As transcription is the first step leading to gene expression (Ananiadou and Mcnaught, 2005), there exist some correlations or associations between the two event types. In tackling this problem, we processed the overlapping rules based on a conditional probability $P(t|E)$, where *t* stands for an event trigger and *E* represents one of the event types. Eq.(1) is used to estimate the value of $P(t_i|E)$.

$$P(t_i|E) = \frac{f(t_i, E)}{\sum_i f(t_i, E)}, \quad (1)$$

where $f(t_i, E)$ is the frequency of the event trigger t_i of the event type E in the training data, and $\sum_i f(t_i, E)$ calculates the total frequency of all event triggers of the event type E in the training data.

$P(t_i|E)$ evaluates the degree of the importance of a trigger to an event type. When the dependency graphs of two rules of different event types are isomorphic to each other, and two rules share a same event trigger, we examine the $P(t_i|E)$ of each event type, and only retain the rule for which the $P(t_i|E)$ is higher.

Compared to the "once a trigger, always a trigger" method employed in other work (Buyko et al., 2009; Kilicoglu and Bergler, 2009), triggers are treated in a more flexible way in our work. A token is not necessarily always a trigger unless it appears in the appropriate context. Also, the same token can serve as trigger for different event types as long as it appears in the different context. A trigger will only be classified into a fixed event type when it could serve as trigger for different event types in the same context.

5.1.3 Performance-based rule ranking

In addition to the process of refining rules across event types, we proposed a performance-based rule ranking method to evaluate each rule under one event type. We matched each rule to sentences in the development set using the subgraph matching approach. For rules that produce at least one event prediction, we ranked them by $PRC(r_i)$, the precision of each rule r_i, which is computed via Eq.(2).

$$PRC(r_i) = \frac{\#correctly\_predicted\_events\_by\_r_i}{\#predicted\_events\_by\_r_i} \quad (2)$$

We manually examined the rules with low rank. In our experiments, the $PRC(r_i)$ ratio of these rules is bigger than 4:1. We removed the ones that are either incorrect or ambiguous in semantics and syntactics based on our domain knowledge. Our assumption is that these rules will keep producing false positive events on the testing data if they are retained in the rule set. For rules that do not make any predictions on the development data, we keep them in the set in the hope that they may contribute to the event recognition from the testing data. Without affecting much on the recall, this process helps to improve the precision of the events extracted from the development data.

5.1.4 GE Results on Development Set

In our previous work (Liu et al., 2010), the matching criteria, "E+P+T" and and "E+P+A", achieved the highest F-score and the highest precision respectively among all the investigated matching criteria. "E+P+T" requires that edge directions and labels of all edges (E) be identical, POS tags (P) of all tokens be identical, and tokens of only event triggers (T) be identical for the edges and the nodes of a rule and a sentence to match with each other. "E+P+A" requires that edges (E), POS tags (P) and all tokens (A) be exactly the same. In this work, we focused on these two criteria and explored to extend them for graph matching between event rules and sentences.

We attempted to relax the matching criterion of POS tags for nouns and verbs. For nouns, the plural form of nouns is allowed to match with the singular form, and proper nouns are allowed to match with regular nouns. For verbs, past tense, present tense and base present form are allowed to match with each other.

Next, letters of each token are transformed into lower case, and tokens containing hyphens are normalized into non-hyphenated forms. Lemmatization is then performed on every pair of tokens to be matched using WordNet (Fellbaum, 1998) as the lemmatizer to allow tokens that share a same lemma to match. Since WordNet is a lexical database only for the general English language, the lemma of a fair amount of domain-specific vocabulary cannot be found in WordNet, such as "Phosphorylation" and "Methylation". In this case, a backup process is invoked to stem the tokens to their root forms using the Porter's stemming algorithm (Porter, 1997) allowing the tokens derived from a same root word to match.

To further generalize event rules, we extended the matching criteria "E+P*+A*" to "E+P*+A*S" to allow tokens to match if their lemmatized forms have a common synonym in terms of the synsets of WordNet. Since WordNet will relate verbs such as "induce" and "receive" together as they share a synonym "have", and allow nouns like "expression" and "aspect" to match as they share a synonym "face", we limited this extension to only adjective tokens to avoid too many false positive events and allow tokens like "crucial" and "critical" to match.

Table 2 shows the event extraction results on the development data based on different matching criteria. The performance is evaluated by "Approximate Span Matching/Approximate Recursive Matching", the primary evaluation measure of the shared task. "E+P*+T*", "E+P*+A*" and "E+P*+A*S" demonstrate the performance of the extended criteria.

Feature	Recall(%)	Prec.(%)	F-score(%)
E+P+A	28.03	66.74	39.48
E+P+T	31.17	52.38	39.09
E+P*+A*	31.45	63.51	42.07
E+P*+T*	35.71	46.26	40.31
E+P*+A*S	31.51	63.32	42.08

Table 2: GE results on development set using different matching criteria

As the strictest matching criteria, "E+P+A" performs better than "E+P+T" in both precision and F-score. Although "E+P+T" achieves a better recall, when relaxing the matching criteria from all tokens being the same to only event trigger tokens having to be identical, the precision of "E+P+T" is decreased by a large margin, nearly 14%. This indicates that a certain number of biological events are described in very similar ways in the literature, involving same grammatical structures and identical contextual contents. While producing more incorrect events, "E+P*+A*" and "E+P*+T*" significantly improve the recall, leading to a better F-score over "E+P+A" and "E+P+T". This confirms the effectiveness of the POS relaxation and the token lemmatization on the generalization of event rules. "E+P*+A*S" obtains a comparable performance with "E+P*+A*" with only a 0.06% increase in recall and a 0.2% drop in precision.

5.1.5 GE Results on Testing Set

Table 3 shows our results of "E+P*+A*" on the testing data using the official metric. We are listed as team "CCP-BTMG". Ranked by F-score, our performance ranked 10th out of 15 participating groups. It

is worth noting that our result on the event type "Protein_catabolism" ranked 1st.

Event type	Rec.(%)	Prec.(%)	F(%)
Gene_expression	58.68	75.77	66.14
Transcription	39.08	51.91	44.59
Protein_catabolism	66.67	83.33	74.07
Phosphorylation	63.78	85.51	73.07
Localization	29.32	91.80	44.44
Binding	22.61	49.12	30.96
Regulation	12.99	46.73	20.33
Positive_regulation	21.90	44.51	29.35
Negative_regulation	15.76	40.18	22.64
All total	31.57	58.99	41.13

Table 3: GE results of "E+P*+A*" on testing set by "Approximate Span /Approximate Recursive Matching"

The performance of our system on the testing set is consistent with that of the development set. We achieved a comparable precision with the top systems and ranked 6th by precision. However, our recall was lower, ranking 11th. This adversely impacted the overall F-score. The lower recall is not surprising because the graph matching criteria "E+P*+A*" strictly demand that every lemmatized token in the patterns, other than protein names represented as"BIO_Entity", has to find its exact match in the input sentences. The detailed analysis on the recall problem is presented in the "Error Classification" section.

While examining the false positives, we found that for many cases our result matched the gold annotation but for the trigger word. We believe that event type and their arguments are more important biologically than the trigger. We consulted some domain experts who reinforced our intuition in many cases that different words could be considered as trigger for the event in question. Following this we contacted organizers and they agreed to release a new evaluation scheme to ignore the trigger match requirement in order to support evaluation of the event extraction itself.

Table 4 shows our results of "E+P*+A*" evaluated by other official evaluation metrics of the task. The strict matching scheme requires exact trigger span as well as all its nested events to be recursively correct for an event to be considered correctly extracted. Our F-score in terms of the strict matching is only 2.65% lower than the relaxed, primary measure, indicating that most of the detected triggers are captured with correct text span. The organizers also provided the evaluation results on PubMed abstracts and PMC full text articles separately. Our system performs consistently on both abstracts and full papers and the difference between F-scores is less than 1% (41.39% vs. 40.47%) mostly due to the small recall loss on full texts.

Measures	R(%)	P(%)	F(%)
Strict Matching	29.55	55.13	38.48
Appr. SpanNoTrigger/Recur.	33.68	62.17	43.69
Appr. Span/Recur./Decomp.	32.56	66.20	43.65
Appr. Sp. No T./Recur./Decomp.	34.96	69.87	46.60
Appr. Span/Recur. (Abstract)	31.87	59.02	41.39
Appr. Span/Recur. (Full paper)	30.82	58.92	40.47

Table 4: GE results on testing set by other evaluation measures

5.2 EPI task

5.2.1 Preprocessing Results

Table 5 presents some statistics of the datasets. We were able to build event rules for 1598 gold events. After removing duplicate rules, we obtained 1,562 event rules distributed over fifteen event types.

Attributes Counted	Training	Dev.	Testing
Abstracts	600	200	440
Total sentences	6,411	2,218	4,640
Candidate sentences	1,054	1,241	2,839
Total events	1,738	582	1,194
Sentence-based events	1,643	536	hidden

Table 5: Statistics of EPI dataset

We processed the obtained rules following the same rule refining and ranking processes of the GE task. We experimented with two graph matching criteria for extracting EPI events, "E+P*+T*" and "E+P*+A*". From the preliminary results, we observed that "E+P*+A*" achieves a high precision over 80% but a lower recall around 33%. Compared to the GE task results, "E+P*+T*" achieves a better recall against a small tradeoff for precision. We consider that this is because the event triggers themselves for the EPI task such as "acetylation", "deglycosylation" and "demethylation" are powerful enough to differentiate among event types without the need to resort to more contextual content of the patterns. Therefore, we focused on using "E+P*+T*" to extract events.

5.2.2 Recall-oriented rule merging

Since all the event types except *Catalysis*, *DNA_methylation* and *DNA_demethylation* in the

EPI task involve addition or removal of biochemical functional groups at a particular amino acid residue of a protein (Hunter, 2009), common syntactic structures of expressing the protein PTM events might be shared across event types. To further improve the recall, we proposed a rule merging strategy to take advantage of the syntactic structures of rules across event types.

We first experimented with a "pairwise flip" approach which combines rules of the pairwise, positive and negative event types by flipping the type and the trigger of event rules. For instance, the event rules of *Phosphorylation* and *Dephosphorylation* are merged together and then used to detect events of the two types respectively.

Next, the "pairwise flip" approach was extended to an "all in one" method. For one event type, the rules of all other PTM event types are processed and merged into the rules of the current type if the trigger of rules of other types contains one of these 12 morphemes: "acetyl", "glycosyl", "hydroxyl", "methyl", "phosphoryl", "ubiqui", "deacetyl", "deglycosyl", "dehydroxyl", "demethyl", "dephosphoryl", "deubiqui". We consider that event rules involving these morphemes in trigger are more likely to discuss representative protein post-translational modifications.

5.2.3 EPI Results on Development Set

Table 6 shows the event extraction results on the development data using different matching criteria and rule merging methods. The performance is evaluated by the primary evaluation measure.

Feature	Recall(%)	Prec.(%)	F(%)
E+P*+A*	32.65	79.83	46.34
E+P*+T*	38.14	73.51	50.23
E+P*+A*(pairwise)	35.22	80.39	48.98
E+P*+T*(pairwise)	40.89	77.52	53.54
E+P*+T*(all in one)	46.39	63.08	53.47

Table 6: EPI results on development set

The two rule merging methods using "E+P*+T*" outperform others in terms of F-score. The "pairwise flip" method achieves higher precision as the syntactic structures of rules to describe the pairwise, positive and negative events tend to be highly similar. However, when merging all the rules across PTM event types, although more events are captured, rules that involve syntactic structures for expressing very specific events of certain types may not generalize well on some other types, resulting in incorrect events. Thus, the "all in one" approach significantly improves the recall while producing many false positive events, leading to a F-score comparable with the "pairwise flip" method.

5.2.4 EPI Results on Testing Set

We conducted two runs on the testing data in terms of "E+P*+T*(pairwise)" and "E+P*+T*(all in one)". Since the two rule merging methods achieve comparable F-scores, we decided to submit a run with higher recall. Table 7 shows our results of "E+P*+T*" using the "all in one" approach on the official metrics. Only 7 teams participated in this task. For the core task, our performance ranked 7th, only 0.16% lower in F-score than the 6th team. When evaluating our results in terms of the full task, we ranked 6th.

Feature	Recall(%)	Prec.(%)	F(%)
E+P*+T*(core task)	45.06	63.37	52.67
E+P*+T*(full task)	23.44	37.93	28.97

Table 7: EPI results on testing set

Compared to the top teams, our F-score is mostly affected by the lower recall. Although the run we submitted achieves the highest recall among all our runs, our recall is about 20% less than the best performing system. Considering that most of the event types of the EPI task tend to use tokens containing only a small fixed set of domain-specific morphemes as triggers, the recall deficit is assumed to be lack of event rules that describe syntactic structures of expressing a fair amount of EPI events.

5.3 Error Classification

Since the gold event annotation of the testing data is hidden, we examined the event extraction results of the development data to analyze the underlying errors. The detailed analysis is reported in terms of false negative and false positive events.

5.3.1 False negatives

It is shown that false negative events have a substantial impact on the performance of all 15 participating teams of the GE task. The best recall, 49.56%, captures less than half of the gold events in the testing set. In our work, three major causes of false negatives are determined for both tasks.

(1) **Low coverage of rule set**: For the GE task, the graph matching criteria "E+P*+A*" strictly asks every lemmatized token in the patterns to find its exact match in the input sentences. Although maintaining the precision at a high level, this directly limits the contextual

structure and content around the proteins and thus prevents the recall from being higher.

Lemmatization helps to detect more events, however, further generalization needs to be performed on the existing rules to relax the token matching requirement. For instance, when "lysine" appears in an event rule, knowing that "lysine" is an amino acid, the rule might be further generalized to allow all amino acids to match with each other in order to recognize more events.

For the EPI task, although "E+P*+T*" requires tokens of only event triggers to be identical, we captured less than half of the gold events. We noticed that many trigger tokens in the development sentences do not appear as triggers in the training set. This leads to the failure of extracting the corresponding events. Since the training data is the only source of triggers in our work, the coverage of triggers limits the generalization power of event rules.

For both tasks, we found that many gold events are described in grammatical structures that are not covered by the existing rules induced from the training sentences. These structures tend to be more complex, involving a long dependency path from the trigger to arguments in the graphs of sentences. Events that consist of these structures are not recognized as no matched rules will be returned from the subgraph matching.

In order to further improve the recall, some postprocessing steps are necessary to be performed on the raw dependency graphs of both rules and sentences instead of using them in the graph matching directly. By eliminating semantically unimportant nodes and grouping lexically connected nodes together, the rules can be generalized to retain only their skeleton structures while complex sentences can be syntactically simplified to allow event rules to match them.

(2) **Compound error effect**: In both tasks, regulation and catalysis event types can take sub-events as arguments. Therefore, if the nested sub-events are not correctly identified, the main events will not be extracted due to the compound error effect.

(3) **Anaphora and coreference**: Since our system focuses on extracting events from sentences, events that contain protein names spanning multiple sentences will not be captured. Recognition of these events requires the ability to do anaphora and coreference resolution in biological text (Gasperin and Briscoe, 2008).

5.4 False positives

Three major causes of false positives are generalized from our analysis.

(1) **Assignment of overlapping event rules**: The conditional probability-based method to assign overlapped rules of different event types effectively reduces the number of event candidates but leads to errors. For instance, "methylation" is used as the trigger for two overlapping rules of *DNA_methylation* and *Methylation*. Based on the $P(t_i|E)$, "methylation" is classified into *DNA_methylation*. An erroneous *DNA_methylation* event is then detected from a development sentence instead of the gold *Methylation* event. Although the trigger and the participant are all identified correctly, the event type is assigned wrongly.

In fact, the same contextual structure and content appear in both *DNA_methylation* and *Methylation* events in the training data. According to the EPI task (Ohta et al., 2011), *Methylation* is to abbreviate for "protein methylation" and thus is different from *DNA_methylation*. In this case, the only way to distinguish between the two types is to identify that the biological entity mentioned in the sentence is a gene for *DNA_methylation* and a protein for *Methylation*. Since genes and their products are uniformly annotated as "Protein" in the task, it is not possible to assign a correct event type in this case from the perspective of the event extraction itself.

(2) **Lack of postprocessing rules**: Some misidentified events require customized postprocessing rules. For instance, a *Gene_expression* event is detected from the phrase "Tax expression vector" of a development sentence. However, since "Tax expression" is only used as an adjective to describe "vector" in this context, the identified *Gene_expression* event is not appropriate. Likewise, "Sp1 transcription" should not be identified as an event in the context of "Sp1 transcription factors".

(4) **Inconsistencies in gold annotation**: Some extracted events are considered biologically meaningful but evaluated as false positives due to the inconsistencies in the gold annotation. In Table 4, the 3.2% increase in precision of the no-trigger evaluation measure over the primary evaluation scheme indicates that the inconsistent gold annotations of event triggers.

6 Conclusion and future work

We used dependency graphs to automatically induce biological event rules from annotated events. We explored methods such as performance-based rule ranking to improve the accuracy of the obtained rules, and we merged rules across multiple event types in order to increase the coverage of the rules. The event extraction process is treated as a subgraph matching problem to

search for the graph of an event rule within the graph of a sentence. We tackled two main tasks of the BioNLP Shared Task 2011. We achieved a 41.13% F-score in detecting events across nine types in the Task 1 of the GE task, and a 52.67% F-score in identifying events across fifteen types in the core task of the EPI task.

In future work, we would like to explore the approaches of generalizing the raw dependency graphs of both event rules and sentences in order to improve the recall of our event extraction system. We also plan to extend our system to tackle the other sub-tasks in GE and EPI tasks, such as to extract events with additional arguments like site and location, and to recognize negations and speculations regarding the extracted events.

References

Muhammad Abulaish and Lipika Dey. 2007. Biological relation extraction and query answering from medline abstracts using ontology-based text mining. *Data & Knowledge Engineering*, 61(2):228–262.

Sophia Ananiadou and John Mcnaught. 2005. *Text Mining for Biology And Biomedicine*. Artech House Publishers.

Sophia Ananiadou, Sampo Pyysalo, Jun'ichi Tsujii, and Douglas B. Kell. 2010. Event extraction for systems biology by text mining the literature. *Trends in Biotechnology*, 28(7):381–390.

Ekaterina Buyko, Erik Faessler, Joachim Wermter, and Udo Hahn. 2009. Event extraction from trimmed dependency graphs. In *BioNLP '09: Proceedings of the Workshop on BioNLP*, pages 19–27, Morristown, NJ, USA. Association for Computational Linguistics.

Luigi P. Cordella, Pasquale Foggia, Carlo Sansone, and Mario Vento. 2004. A (sub)graph isomorphism algorithm for matching large graphs. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(10):1367–1372.

Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. 2001. *Introduction to Algorithms*. The MIT Press.

Christiane Fellbaum. 1998. *WordNet: An Electronic Lexical Database*. Bradford Books.

Caroline Gasperin and Ted Briscoe. 2008. Statistical anaphora resolution in biomedical texts. In *COLING '08: Proceedings of the 22nd International Conference on Computational Linguistics*, pages 257–264, Morristown, NJ, USA. Association for Computational Linguistics.

Gérard P. Huet. 1975. A unification algorithm for typed lambda-calculus. *Theor. Comput. Sci.*, 1(1):27–57.

Lawrence Hunter. 2009. *The Processes of Life: An Introduction to Molecular Biology*. The MIT Press.

Halil Kilicoglu and Sabine Bergler. 2009. Syntactic dependency based heuristics for biological event extraction. In *Proceedings of the Workshop on Current Trends in Biomedical Natural Language Processing: Shared Task*, BioNLP '09, pages 119–127.

Jin-Dong Kim, Yoshinobu Kano Tomoko Ohta, Sampo Pyysalo, and Jun'ichi Tsujii. 2009. Overview of bionlp'09 shared task on event extraction. In *Proceedings of the NAACL-HLT 2009 Workshop on Natural Language Processing in Biomedicine (BioNLP'09)*, pages 1–9. ACL.

Jin-Dong Kim, Sampo Pyysalo, Tomoko Ohta, Robert Bossy, and Jun'ichi Tsujii. 2011a. Overview of BioNLP Shared Task 2011. In *Proceedings of the BioNLP 2011 Workshop Companion Volume for Shared Task*, Portland, Oregon, June. Association for Computational Linguistics.

Jin-Dong Kim, Yue Wang, Toshihisa Takagi, and Akinori Yonezawa. 2011b. Overview of the Genia Event task in BioNLP Shared Task 2011. In *Proceedings of the BioNLP 2011 Workshop Companion Volume for Shared Task*, Portland, Oregon, June. Association for Computational Linguistics.

Haibin Liu, Vlado Keselj, and Christian Blouin. 2010. Biological event extraction using subgraph matching. In *Proceedings of the 4th International Symposium on Semantic Mining in Biomedicine (SMBM-2010)*, October.

Yusuke Miyao, Kenji Sagae, Rune Saetre, Takuya Matsuzaki, and Jun'ichi Tsujii. 2009. Evaluating contributions of natural language parsers to protein–protein interaction extraction. *Bioinformatics*, 25(3):394–400.

Tomoko Ohta, Yuka Tateisi, and Junichi Tsujii. 2005. Syntax annotation for the genia corpus. In *Proceedings of the IJCNLP 2005*, pages 222–227.

Tomoko Ohta, Sampo Pyysalo, and Jun'ichi Tsujii. 2011. Overview of the Epigenetics and Post-translational Modifications (EPI) task of BioNLP Shared Task 2011. In *Proceedings of the BioNLP 2011 Workshop Companion Volume for Shared Task*, Portland, Oregon, June. Association for Computational Linguistics.

Mate Ongenaert, Leander Van Neste, Tim De Meyer, Gerben Menschaert, Sofie Bekaert, and Wim Van Criekinge. 2007. Pubmeth: a cancer methylation database combining text-mining and expert annotation. *Nucleic Acids Research*, pages 1–5.

Marcello Pelillo, Kaleem Siddiqi, and Steven W. Zucker. 1999. Matching hierarchical structures using association graphs. *IEEE Trans. Pattern Anal. Mach. Intell.*, 21(11):1105–1120.

M. F. Porter. 1997. An algorithm for suffix stripping. pages 313–316.

Yuanyuan Tian, Richard C. Mceachin, Carlos Santos, David J. States, and Jignesh M. Patel. 2007. Saga: a subgraph matching tool for biological graphs. *Bioinformatics*, 23(2):232–239.

J. R. Ullmann. 1976. An algorithm for subgraph isomorphism. *J. ACM*, 23(1):31–42.

Xifeng Yan, Feida Zhu, Jiawei Han, and Philip S. Yu. 2006. Searching substructures with superimposed distance. In *ICDE '06: Proceedings of the 22nd International Conference on Data Engineering*, page 88, Washington, DC, USA. IEEE Computer Society.