Biological Event Extraction using Subgraph Matching

Haibin Liu
Faculty of Computer Science
Dalhousie University
Halifax, NS, Canada
haibin@cs.dal.ca

Christian Blouin
Faculty of Computer Science
Dalhousie University
Halifax, NS, Canada
cblouin@cs.dal.ca

Vlado Kešelj
Faculty of Computer Science
Dalhousie University
Halifax, NS, Canada
vlado@cs.dal.ca

Abstract

An important task in biological information extraction is to identify descriptions of biological relations and events involving genes or proteins. We propose a graph-based approach to automatically learn rules for detecting biological events in the literature. The detection is performed by searching for isomorphism between event rules and the dependency graphs of complete sentences. When applying our approach to the datasets of the Task 1 of the BioNLP shared task, we achieved an 37.28% F-score in detecting biological events across 9 event types.

1 Introduction

Recent research in information extraction in the biological domain has focused on extracting semantic relations between molecular biology concepts (Fundel et al., 2007). State-of-the-art protein annotation methods have achieved reasonable success with a performance of 88% F-score (Wilbur et al., 2007). A task of interest is to automatically extract protein-protein interactions (PPI). To date, most of the biological knowledge about these interactions is only available in the form of unstructured text from scientific articles (Abulaish and Dey, 2007). The best-performing system from the BioCreative II challenge (Hunter et al., 2008) only achieved a 29% F-score in identifying protein pairs in a sentence that have a biologically relevant relationship. This suggests that the problem of biological relation extraction is difficult and far from solved.

Sentences in the biological literature often have long-range dependencies. Therefore, co-occurrence based or surface pattern based shallow analysis on biological texts suffers from either low precision or recall (Fundel et al., 2007; Abulaish and Dey, 2007). As a result, full parsing has been explored as the basis for relation extraction to perform intensive syntactical and semantic analysis (Abulaish and Dey, 2007; Fundel et al., 2007; Rinaldi et al., 2007). In the BioNLP'09 shared task on biological event extraction (Kim et al., 2009), 20 out of the total 24 participating teams resorted to a full parsing strategy, including all top 10 performing teams. However, most of previous work extracts relevant relations based on a limited set of manually designed rules that map interpreted syntactic structures into the semantic relations. We propose an approach to automatically learn rules that characterize a wide range of biological relations and events from a syntactically and semantically annotated corpus, and our approach is also based on full parsing of biological texts.

More recently, the dependency representation obtained from full 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). Relations are generally extracted from the dependency representation by two approaches. In one approach, the dependency representation is traversed and paths that contain the relevant terms describing the relations predefined in the rules are extracted as candidate relations (Fundel et al., 2007; Rinaldi et al., 2004). In the other, relations are learned from the dependency representation using supervised machine learning based on specialized feature representations or kernels, encoded with dependency paths from the representation (Airola et al., 2008; Björne et al., 2009).

Graphs provide a powerful primitive for modeling biological data such as pathways and protein interaction networks (Tian et al., 2007; Yan et al., 2006). Since the dependency representation maps straightforwardly onto a directed graph (de Marneffe and Manning, 2008), properties and operations of graphs can be naturally applied to the problem of biological relation extraction. We propose a graph matching-based approach to extract biological events from the scientific literature in tackling the primary task of the BioNLP'09 shared task on biological event extraction. The extraction is performed by matching the dependency representation of automatically learned rules to the de-

pendency representation of biological sentences. 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.

The rest of the paper is organized as follows: In Section 2, we introduce the BioNLP'09 shared task on event extraction. Section 3 describes our subgraph matching-based event extraction method. Sections 4 elaborates the implementation details. Performance is evaluated in Section 5. Finally, Section 6 summarizes the paper and introduces future work.

2 BioNLP'09 Shared Task

The BioNLP'09 shared task (Kim et al., 2009) focused on the recognition of biological events that appear in the biological literature. When a biological event is described in text, we can analyze it by recognizing an *event type*, the *event trigger*, one or more *event arguments*, and the *source text* (ST), where the event is described. The source text is composed of *tokens*, which are defined as finite strings of characters from a finite alphabet. The *alphabet* is a finite set of symbols Σ. Tokens come from W, the set of all finite strings of characters from Σ, i.e., $W = \Sigma^+$. The source text is a finite sequence of tokens, i.e., any member of W^*. We define a *biological event* in a way consistent with the shared task, which is as follows:

Definition 1. A biological event is a four-tuple $e =$ (*Type*, *Trigger*, *Arguments*, *ST*). $ST \in W^*$, called the *source text*, is a sequence of tokens that contains the event; *Type* $\in T_e$ is an event type from a finite set of event types T_e; *Trigger* is a substring of tokens from *ST* that signals the event; *Arguments* is a non-empty, finite set of pairs (l, a) where $l \in L$ is a label from a finite set of semantic role labels L, and a is a token from *ST*, or another biological event.

For the shared task, T_e consists of nine event types defined in Table 1, and $L = \{\text{Theme, Cause}\}$. A *gold event* denotes a biological event where all the information has been manually annotated by domain experts.

The primary task of the shared task was to detect biological events such as protein binding and phosphorylation, given only 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. We focus on the primary task and propose a graph matching-based method to cope with the problem.

3 Subgraph Matching-based Event Extraction

3.1 Dependency Representation

The dependency representation is designed to provide a simple description of the grammatical relationships in a sentence that can be effectively used to extract textual relations (de Marneffe and Manning, 2008).

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

3.2 Event Rule Induction

A *biological event rule* is defined as follows:

Definition 2. A biological event rule is a pair $r = (e, G_r)$. $G_r = (V_r, E_r)$ is a dependency graph, which characterizes the contextual structure of events. $e =$ (*Type*, *Trigger*, *Arguments*) encodes a detailed event frame, where *Type* is the event type, *Trigger* = $\{(t_1, v_1), (t_2, v_2), \cdots\}$ records the event trigger and is a non-empty finite sequence of tokens associated with nodes in G_r, i.e., *Trigger* $\in (W \times V_r)^+$, and *Arguments* = $\{(t_1, l_1, v_1), (t_2, l_2, v_2), \cdots\}$ records the event arguments and is a non-empty finite sequence of tokens associated with semantic role labels and nodes in G_r, i.e., *Arguments* $\in (W \times L \times V_r)^+$.

The biological event rules are learned from labeled training sentences using the following induction method. Starting with the dependency graph of each training sentence, the directions of edges 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 for each event, 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 obtained rules are categorized in terms of the nine event types of the task.

3.3 Sentence Matching

We propose a sentence matching approach to attempt to match event rules to each testing sentence. 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. This problem is also called *subgraph isomorphism*, defined in this work as follows:

Definition 3. 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). Considering that the graphs of rules and sentences involved in our matching process are small, a simple subgraph matching algorithm using a backtracking approach is appropriate. It is named "Injective Graph Embedding Algorithm" and designed based on the Huet's graph unification algorithm (Huet, 1975). The main and the recursive part of the algorithm are formalized in Algorithm 1 and Algorithm 2.

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 such as the type to the corresponding tokens of the sentence. In practice, it only takes the algorithm a couple of seconds to return the results.

4 Implementation

We assume that a sentence is a suitable level of text granularity in event extraction. The target text is first segmented into sentences. Then, each sentence is tokenized with whitespace separating tokens. We require that every protein be separated from surrounding text and become one individual token. All the protein names are replaced with a unified tag "BIO_Entity".

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 biological event rules $R = \{r_1, r_2, \cdots, r_i, \cdots\}$, 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 biological event rules from R matched with s together with the injective mapping

Main algorithm:

```
MR ← ∅
for all r_i ∈ R do
    st_{r_i} ← StartNode(G_{r_i})        //StartNode finds the start
    //node st_{r_i} of the rule graph G_{r_i}
    ST_s ← {st_{s_1}, st_{s_2}, ···, st_{s_j}, ···}
    //ST_s : the set of start nodes of the sentence graph G_s
    for all st_{s_j} ∈ ST_s do
        create an empty stack σ and push (st_{r_i}, st_{s_j}) onto
        the stack σ
        IM ← ∅                        //IM : records of injective matches
        //between nodes in G_{r_i} and G_s
        call MatchNode(σ, rIM, G_{r_i}, G_s)
        //rIM : reference of IM
        if MatchNode returned TRUE then
            MR ← MR ∪ {r_i with IM }
return MR
```

GENIA tagger (Tsuruoka et al., 2005) is used to associate each word in the tokenized sentences with its most likely Part-of-Speech tag. The POS-tagged sentences are submitted to the Stanford unlexicalized natural language parser (Klein and Manning, 2003) to analyze the syntactic and semantic structure of the sentences. The Stanford parser returns a dependency graph for each sentence after parsing.

For each gold event, the shortest path in the undirected graph connecting the event trigger to each event argument is extracted using the Dijkstra's algorithm (Cormen et al., 2001) with equal weight for edges. Sentence matching is performed following the procedure of Algorithm 1 and Algorithm 2.

5 Results and Evaluation

5.1 Dataset

We use the BioNLP'09 Shared Task datasets for evaluation. A training set and a development set are provided for the purpose of developing task solution. They are prepared based on the publicly available portion of the GENIA event corpus (Kim et al., 2008) with the gold protein annotation and the gold event annotation given. A testing set is prepared from a held-out part of the corpus and provided without the gold event annotation.

Table 1 shows the nine event types considered in the

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 σ is not empty **do**
4: pop node pair (v_r, v_s) from stack σ
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 a same direction and
19: possess identical edge labels **then**
20: $S \leftarrow S \cup n_s$ //S : the set of candidate
21: //nodes for matching n_r
22: **for all** $n_s \in S$ **do**
23: **if** an injective match between n_r and n_s
24: already exists in $IM_{current}$ **then**
25: go to Line 14 and proceed with next edge e_r
26: **else if** an injective match is possible between
27: n_r and n_s **then**
28: $\sigma_n \leftarrow \sigma$ //copy σ to a new stack σ_n
29: push (v_r, v_s, n_r, n_s) onto the stack σ_n
30: **call** MatchNode($\sigma_n, rIM_{current}, G_{r_i}, G_s$)
31: //$rIM_{current}$: reference of $IM_{current}$
32: **if** MatchNode **returned** TRUE **then**
33: $IM_{parent} \leftarrow IM_{current}$
34: //update IM_{parent} using $IM_{current}$
35: **return** TRUE
36: **return** FALSE
37: $IM_{parent} \leftarrow IM_{current}$
38: **return** TRUE

shared task. Since these types are all related to protein biology, they take proteins (P) as their theme. Regulation events always take a theme argument and, when expressed, also a cause argument. As a unique feature of the shared task, regulation events may take another event (E), namely sub-event, as its theme or cause.

5.2 Rule Induction 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 proposed graph matching-based method focuses on extracting biological events from sentences. Therefore, only sentence-based events are considered in this work. After removing duplicate rules, we obtained 6,435 event rules, which are distributed over nine event types.

	Event type	Primary arguments
1	Gene_expression	Theme(P)
2	Transcription	Theme(P)
3	Protein_catabolism	Theme(P)
4	Phosphorylation	Theme(P)
5	Localization	Theme(P)
6	Binding	(Theme(P))$^+$
7	Regulation	Theme(P/E), (Cause(P/E))$^?$
8	Positive_regulation	Theme(P/E), (Cause(P/E))$^?$
9	Negative_regulation	Theme(P/E), (Cause(P/E))$^?$

Table 1: Event types and primary arguments

We observed that some event rules of an event type are overlapped 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.

In tackling this problem, we processed the rules and built a non-overlapping rule set. When the dependency graphs of two rules across different event types are isomorphic to each other and two rules share a same event trigger token, we keep the rule of the event type in which the trigger token of the rule occurs more frequent as a trigger in the training data, and remove the rule of the other event type from the set.

5.3 Event Extraction Results on Development Set

The non-overlapping rule sets in terms of different combinations of matching features are then applied to the 988 candidate development sentences using our graph matching algorithm. Table 2 shows the event extraction results based on each feature.

The least specific matching criterion when matching between rules and sentences is "E", which assumes that, without checking any information about nodes, as long as edge directions and labels are the same, both edges and nodes of a rule and a sentence can match with each other. It achieves the highest recall among all the runs and captures more than half of the gold events in the sentences. However, the precision is quite low, leading to a low F-score as too many false positives are generated due to the disregard of node information.

As the strictest matching criteria, "E+P+A" requires

that the edges (E), the POS tags (P) and all tokens (A) be exactly the same for the edges and the nodes of a rule and a sentence to match with each other. It achieves the highest precision 69.72% and an F-score over 40%. This indicates that a certain number of biological events are described in very similar way in the literature, involving the same grammatical structures and identical contextual contents. Comparing to "P+A", adding the edge features improves the overall precision of event extraction by a large margin, nearly 13%. "E+P+T" requires that edge directions and labels of all edges be identical, POS tags of all tokens be identical, and tokens of only event triggers (T) be identical. It achieves better performance than "E+P+A" when relaxing the matching criteria from all tokens being the same to only event trigger tokens having to be identical. The best 2 of the first 6 runs in Table 2 are "E+P+T" and "P+A".

Next, 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. Further, the event trigger tokens are stemmed to their root forms allowing the trigger tokens derived from a same root word to match. "E+P*+T*" and "P*+A+T*" in Table 2 demonstrate the improved performance to the above best two runs. These modifications improve the recall but produce many incorrect events, leading to only a small increase on the overall F-score.

Feature	Prec.(%)	Recall(%)	F-score(%)
E	1.22	**52.26**	2.38
E+P	2.23	45.33	4.25
E+P+A	**69.72**	28.06	40.02
E+P+T	58.85	31.02	**40.63**
P+A	57.00	32.53	**41.42**
P+T	40.65	36.95	38.71
E+P*+T*	50.86	34.71	**41.26**
P*+A+T*	51.51	35.22	**41.84**

Table 2: Event extraction of non-overlapping set on development set using different features

5.4 Event Extraction Results on Testing Set

We decided to conduct four runs on the testing sentences in terms of 4 features: "E", "E+P+A", "E+P*+T*" and "P*+A+T*". For "E" and "E+P+A", aiming to investigate the highest recall and precision on the testing sentences that can be achieved by our method. Table 3 gives the event extraction results on the 1,670 testing sentences in terms of the 4 features.

Feature	Prec.(%)	Recall(%)	F-score(%)
E	0.84	**52.17**	1.65
E+P+A	**58.64**	26.02	36.05
E+P*+T*	41.77	33.66	**37.28**
P*+A+T*	39.61	32.18	35.51

Table 3: Event extraction of non-overlapping set on testing sentences using different features

"E+P*+T*" achieves the best overall F-score of 37.28% among all the runs. Similarly to the development set, the highest precision 58.64% on the testing sentences is achieved by the strictest matching criteria "E+P+A". The highest recall 52.17% is obtained by the least specific matching criterion "E", indicating that a large amount of biological events is described in quite different grammatical structures in the literature. Although "P*+A+T*" produced the best performance on the development set, it does not perform as well on the testing set. This clearly suggests that when requiring every token to be exactly the same for matching nodes of a rule and a sentence, the event rules have less stable generalization power to capture the underlying events.

Table 4 gives the performance comparison of our method with top-performing teams in the task. The official evaluation shows that our best results would rank 6th in extracting biological events in the testing data compared to the results of the 24 participating teams.

Team	Prec.(%)	Recall(%)	F-score(%)
UTurku	58.48	46.73	51.95
JULIELab	47.52	45.82	46.66
ConcordU	61.59	34.98	44.62
UT+DBCLS	55.59	36.90	44.35
VIBGhent	51.55	33.41	40.54
DalhousieU	**41.77**	**33.66**	**37.28**
UTokyo	53.56	28.13	36.88
UNSW	45.78	28.22	34.92

Table 4: Performance comparison with participating teams

6 Conclusion and future work

We use dependency graphs to automatically induce biological event rules from annotated events. These rules are then used to extract biological events from the literature. The extraction process is treated as a subgraph matching problem to search for the graph of an event rule within the dependency graph of a sentence. We

conducted the experiments to tackle the primary task of the BioNLP shared task, and our method achieves an 37.28% F-score on the testing data in detecting biological events across nine event types.

In future work, we would like to experiment with more matching criteria when mapping event rules to sentences. We also plan to expand the coverage of event trigger tokens using external lexical resources for new event triggers and synonyms of existing triggers.

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.

Antti Airola, Sampo Pyysalo, Jari Björne, Tapio Pahikkala, Filip Ginter, and Tapio Salakoski1. 2008. All-paths graph kernel for protein-protein interaction extraction with evaluation of cross-corpus learning. *BMC Bioinformatics*, 9 Suppl 11:s2.

Jari Björne, Juho Heimonen, Filip Ginter, Antti Airola, Tapio Pahikkala, and Tapio Salakoski. 2009. Extracting complex biological events with rich graph-based feature sets. In *BioNLP '09: Proceedings of the Workshop on BioNLP*, pages 10–18, Morristown, NJ, USA. Association for Computational Linguistics.

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

Marie-Catherine de Marneffe and Christopher D. Manning. 2008. The Stanford typed dependencies representation. In *CrossParser '08: Coling 2008: Proceedings of the workshop on Cross-Framework and Cross-Domain Parser Evaluation*, pages 1–8, Morristown, NJ, USA. Association for Computational Linguistics.

Katrin Fundel, Robert Küffner, and Ralf Zimmer. 2007. Relex—relation extraction using dependency parse trees. *Bioinformatics*, 23(3):365–371.

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

Lawrence Hunter, Zhiyong Lu, James Firby, William A. Baumgartner Jr., Helen L. Johnson, Philip V. Ogren, and K. Bretonnel Cohen. 2008. Opendmap: An open source, ontology-driven concept analysis engine, with applications to capturing knowledge regarding protein transport, protein interactions and cell-type-specific gene expression. *BMC Bioinformatics*, 9.

Jin-Dong Kim, Tomoko Ohta, and Jun ichi Tsujii. 2008. Corpus annotation for mining biomedical events from literature. *BMC Bioinformatics*, 9(1):10.

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.

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

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.

Fabio Rinaldi, Gerold Schneider, Kaarel Kaljurand, James Dowdall, Christos Andronis, Andreas Persidis, and Ourania Konstanti. 2004. Mining relations in the GENIA corpus. In *Proceedings of the Second European Workshop on Data Mining and Text Mining for Bioinformatics*, Pisa, Italy.

Fabio Rinaldi, Gerold Schneider, Kaarel Kaljurand, Michael Hess, Christos Andronis, Ourania Konstandi, and Andreas Persidis. 2007. Mining of relations between proteins over biomedical scientific literature using a deep-linguistic approach. *Artif. Intell. Med.*, 39(2):127–136.

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.

Yoshimasa Tsuruoka, Yuka Tateishi, Jin-Dong Kim, Tomoko Ohta, John McNaught, Sophia Ananiadou, and Jun ichi Tsujii. 2005. Developing a Robust Part-of-Speech Tagger for Biomedical Text. LNCS 3746:382–392.

John Wilbur, Lawrence Smith, and Lorraine Tanabe. 2007. Biocreative 2. gene mention task. In *Proceedings of Second BioCreative Challenge Evaluation Workshop*, pages 7–16.

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.