J Inf Process Syst, Vol.11, No.2, pp.248~265, June 2015
http://dx.doi.org/10.3745/JIPS.04.0008

ISSN 1976-913X (Print)
ISSN 2092-805X (Electronic)

JOURNAL OF INFORMATION PROCESSING SYSTEMS JIPS

A Maximum Entropy-Based Bio-Molecular Event Extraction Model that Considers Event Generation

Hyoung-Gyu Lee*, So-Young Park**, Hae-Chang Rim*,
Do-Gil Lee***, and Hong-Woo Chun****

Abstract

In this paper, we propose a maximum entropy-based model, which can mathematically explain the bio-molecular event extraction problem. The proposed model generates an event table, which can represent the relationship between an event trigger and its arguments. The complex sentences with distinctive event structures can be also represented by the event table. Previous approaches intuitively designed a pipeline system, which sequentially performs trigger detection and arguments recognition, and thus, did not clearly explain the relationship between identified triggers and arguments. On the other hand, the proposed model generates an event table that can represent triggers, their arguments, and their relationships. The desired events can be easily extracted from the event table. Experimental results show that the proposed model can cover 91.36% of events in the training dataset and that it can achieve a 50.44% recall in the test dataset by using the event table.

Keywords

Bioinformatics, Event Extraction, Maximum Entropy, Text-Mining

1. Introduction

To analyze biomedical literature, some previous approaches have focused only on recognizing named entities (such as proteins), while some recent approaches have emphasized the problem of identifying the interaction between two entities [1-6]. They are interested in extracting binary relations, such as protein-protein interactions and disease-gene associations. However, such binary relations do not provide a deep analysis of biomedical phenomena. Consequently, a bio-event extraction task is required to recognize bio-molecular events that describe a change in the state of the bio-molecular event [7].

For this task, we tried to identify a set of events where each event consisted of a trigger and its arguments [7]. In the example of Fig. 1, a set {($event_1$), ($event_2$)} is recognized and the trigger is identified as *promotes*, and its arguments, such as theme and cause, are also identified. We assumed that the biomedical text was already analyzed with a named entity recognizer, which is a part-of-speech tagger, and a dependency parser, as shown in the lower portion of Fig. 1.

Manuscript received November 12, 2013; accepted December 23, 2013; onlinefirst October 7, 2014.
Corresponding Author: So-Young Park (ssoya@smu.ac.kr)
* Dept. of Computer Science, Korea University, Seoul 136-701, Korea (hglee@nlp.korea.ac.kr, rim@nlp.korea.ac.kr)
** Dept. of Game Design & Development, SangMyung University, Seoul 110-743, Korea (ssoya@smu.ac.kr)
*** Research Institute of Korean Studies, Korea University, Seoul 136-701, Korea (motdg@korea.ac.kr)
**** Technology Information Analysis Center, KISTI, Seoul 130-741, Korea (hw.chun@kisti.re.kr)

The difficulties of bio-molecular event extraction are shown in Fig. 2. In this figure, only Fig. 2(a) is the correct event extraction from among many other possible candidates. One of the difficult event extractions is the case when an event can take other events as its argument [8,9]. For example, ($event_2$) takes ($event_1$) as its *theme* argument, as shown in Fig. 2(a). Furthermore, a correct trigger can be missed, such as in Fig. 2(b) and (f) without the gene expression (GE) trigger *production*. On the opposite hand, an incorrect trigger can be detected, such as in the trigger *activity* shown in (d). Even if all of the correct triggers are detected, there is the chance that an argument cannot be detected, or that the argument type will be incorrectly identified. Compared with the correct event ($event_2$) at (a), for example, the incorrect event ($event_4$) at (c) takes the incorrect *theme* argument and no correct *cause* argument. Even if the correct events are detected, some incorrect events can be unnecessarily detected, as in the event ($event_6$) at (e).

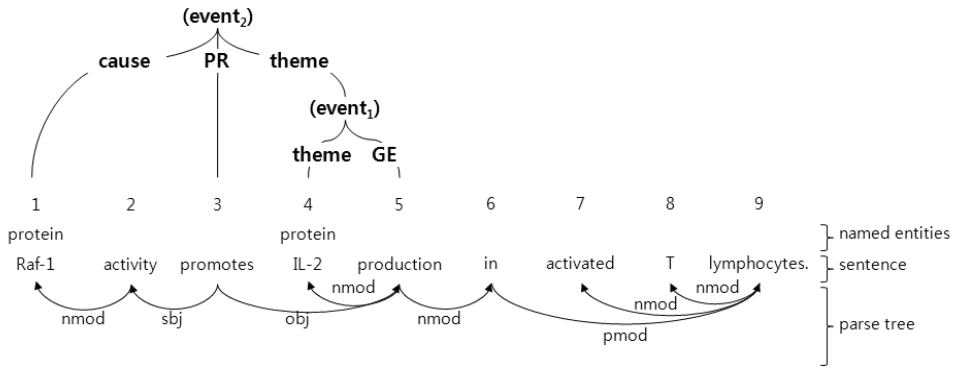


Fig. 1. Bio-molecular event examples extracted from an analyzed sentence.

In this paper, we propose a model for bio-molecular event extraction that estimates the probabilities for generating all possible sets of bio-molecular events from a sentence, and that selects the best event set with the highest probability value. The remainder of this paper is organized as follows: Section 2 surveys some previous approaches, and Section 3 explains the proposed model for bio-molecular event extraction. Then, in Section 4, we demonstrate the experimental results, and the characteristics of the proposed model conclude the paper in Section 5.

2. Previous Work

For bio-event extraction, most approaches first detect the triggers in a sentence, and then they obtain the edges that represent the relationship between a trigger and its arguments [7]. Also, they actively utilize dependency parsing information to detect the edges. This is because several previous approaches have already improved their performance by using features extracted from dependency parsing information [4,5,10-12]. Furthermore, the distance between an event trigger and its arguments tends to be much shorter in the dependency path than in the sentence [8]. On the other hand, they can be classified into rule-based approaches, machine learning based approaches, and dictionary and machine learning based approaches.

First, the rule-based approaches automatically draw out some draft event extraction rules from a training set, and then refine these rules that are defined by experts [13-15]. These accurate event

extraction rules allow for the rule-based approaches to indicate a comparatively high-precision value. They are very superior to other approaches for simple events. However, these approaches cannot guarantee a reasonable recall on difficult events, including binding and regulations. Additionally, the accuracy can be overly dependent on the expert's ability. Therefore, modifying the refined rules and changing the features used for constructing the draft rules is a very difficult task.

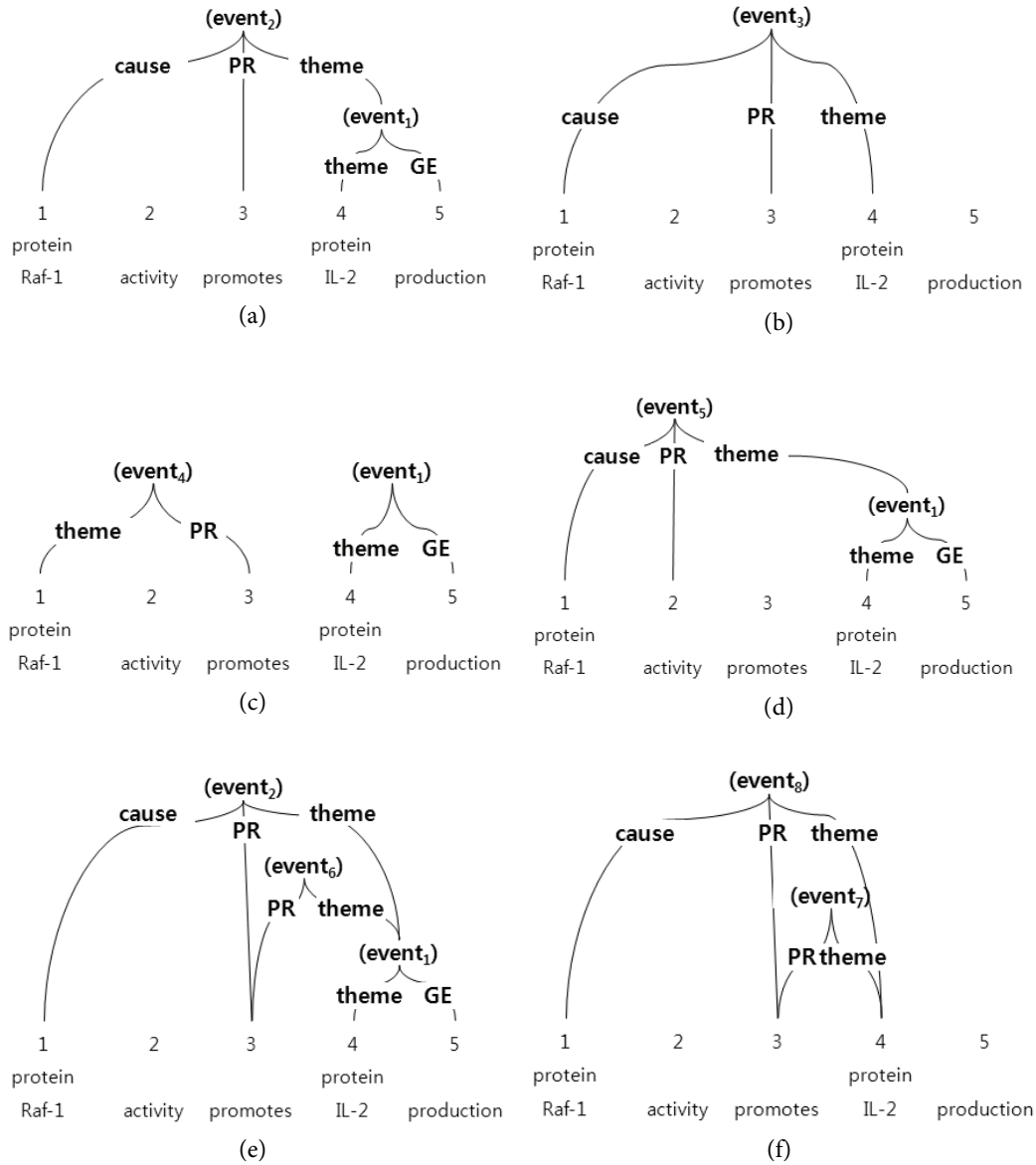


Fig. 2. Candidate event sets extracted from the same sentence.

Second, the machine learning based approaches focus on assigning an event type to an individual token or recognizing an individual relation between a trigger candidate and its argument candidate

[8,16,17]. However, the approaches do not mathematically describe how to decompose the problem of extracting the events from a document into the problem of recognizing the individual trigger and the problem of detecting the arguments. Even though these sub problems can use the same machine learning technique with similar features, the approaches also cannot explain the relationship.

Third, the dictionary and machine learning based approaches use a dictionary for the trigger detection, and a machine learning technique for the argument recognition [9,18,19]. However, these approaches are expensive when it comes to building the dictionary, because the dictionary requires the expert's effort.

In this paper, we propose a model to clearly explain the characteristics of the bio-molecular event extraction problem by using mathematical modeling. In order to clearly describe the connection between the trigger detection step and the argument recognition step, our proposed model changes the event extraction problem into the problem of generating an event table, which includes both unary entries for triggers and binary entries for arguments. For the purpose of significantly simplifying the process of solving an event extraction problem by focusing only on the binary relationship between an event trigger and each of its arguments, the proposed model converts the event table generation problem into the problem of generating each entry in the event table. The proposed model is learned from a training set without using the expert's support.

3. The Proposed Bio-Molecular Event Extraction Model

In this chapter, we propose a maximum entropy-based model for bio-molecular event extraction. As shown in Fig. 3, the proposed model consists of a preprocessing step, while analyzing the given document by using natural language processing tools, such as a stemmer, a part-of-speech tagger, and a dependency parser. As the first step in the proposed model, the generation step generates an event table, instead of generating the events themselves. This is done for the purpose of reducing the complexity of solving the event extraction problem, by focusing only on the binary relationship between an event trigger and its argument. Then, the desired events can be easily extracted from the event table. Section 3.1 defines how to estimate the probabilities of generating every entry in the event table. Section 3.3 illustrates the relationship between the event table and a set of the events.

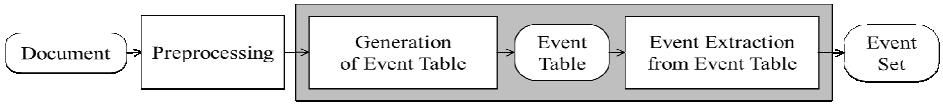


Fig. 3. The proposed bio-molecular event extraction model.

3.1 Generation of the Event Table

The proposed model estimates the probabilities of generating the sets of bio-molecular events from the document Doc^{NE}, and selects the best event set E with the highest probability, as represented on the left hand side of Eq. (1). NLP tools, such as a stemmer, a part-of-speech tagger, and a dependency parser, analyze the document Doc^{NE}. In order to simplify the event extraction problem, some equations are derived in the following way: first, the document Doc^{NE} is divided into two sentence sets of S^{ϕ}, which consist of sentences without any named entity; and S^{NE}, which consists of sentences

with named entities, as described on the right hand side of Eq. (1).

$$\begin{aligned}
&\underset{\mathbb{E}}{argmax} P(\mathbb{E} \mid Doc^{NE}) \\
&= \underset{\mathbb{E}}{argmax} P(\mathbb{E} \mid S^{NE}, S^{\Phi}) && (1)\\
&\approx \underset{\mathbb{E}}{argmax} P(\mathbb{E} \mid S^{NE}) && (2)\\
&= \underset{E_{1m}}{argmax} P(E_{1m} \mid S_{1m}^{NE}) && (3)\\
&= \underset{E_{1m}}{argmax} P(E_1 \mid S_{1m}^{NE}) \times P(E_2 \mid S_{1m}^{NE}, E_1) \times \ldots \times P(E_m \mid S_{1m}^{NE}, E_{1m-1}) && (4)\\
&\approx \underset{E_{1m}}{argmax} \prod_{i=1}^{m} P(E_i \mid S_i^{NE}) && (5)
\end{aligned}$$

Because an unnamed entity indicates that there is no event, the sentence set S^{ϕ} is removed as presented in Eq. (2). Furthermore, the sentence set S^{NE} and the event set E are replaced with the sentence sequence S_{1m}^{NE}, and the event set sequence E_{1m}, as shown in Eq. (3). Also, Eq. (4) generalizes multiple events by the use of the chain rule. Finally, Eq. (5) is simplified with the assumption that the event set of each sentence does not depend on other sentences. For better understanding, Table 1 describes each term's meaning.

$$\begin{aligned}
&\underset{E}{argmax} P(E \mid S^{NE}) \\
&\approx \underset{e_{1,1\ldots}e_{1,n}}{argmax} P(e_{1,1\ldots}e_{1,n} \mid S^{NE}) && (6)\\
&= \underset{e_{1,1\ldots}e_{1,n}}{argmax} \prod_{i=0}^{n-1} \prod_{j=i}^{n-i} P(e_{j,j+i} \mid S^{NE}, e_{history}) && (7)\\
&= \underset{e_{1,1\ldots}e_{1,n}}{argmax} \prod_{j=1}^{n} P(e_{j,j} \mid S^{NE}, e_{history}) \times \prod_{i=1}^{n-1} \prod_{j=1}^{n-i} P(e_{j,j+i} \mid S^{NE}, e_{history}) && (8)
\end{aligned}$$

In order to extract events from an arbitrary sentence by freeing oneself from the position in the sequence S_{1m}^{NE}, the left hand side of Eq. (6) is derived from Eq. (5). In particular, we assume that the event set E can be represented as the proposed event table, as described in both Eq. (6) and Fig. 4. Beginning with the entries that represent the trigger-argument relationship between two near words, Eq. (7) also generalizes multiple events by use of the chain rule. Furthermore, Eq. (8) describes that the event table generation problem can be divided into the trigger generation problem and the trigger-argument relation generation problem. Considering the fast processing time and low memory requirement, the proposed model uses the best-first strategy [20,21] while searching through the sequence of entries.

Table 1. Description of terms

Term	Description
Doc^{NE}	A named entity annotated document, which is analyzed by NLP tools
S^{Φ}	A set of sentences without any named entity in the document Doc^{NE}
S^{NE}	A set of sentences with named entities in the document Doc^{NE} where $S^{NE} \cap S^{\Phi} = \phi$
S_{1m}^{NE}	$S_1^{NE}, S_2^{NE}, \ldots, S_m^{NE}$: a sequence of the sentences in the set S^{NE} where m indicates the number of the sentences.
S_i^{NE}	The i-th sentence in the sequence S_{1m}^{NE}
S^{NE}	A named entity annotated sentence, which is analyzed by NLP tools
E	A set of bio-molecular events that occurred in the document Doc^{NE}
E_i	A set of bio-molecular events that occurred in the sentence S_i^{NE}
E_{1m}	$E_1, E_2, \ldots, E_m$: a sequence of the event sets where $E = E_1 \cup E_2 \cup \ldots \cup E_m$ and $\forall_i \forall_j E_i \cap E_j = \phi$
E	A set of bio-molecular events that occurred in the sentence S^{NE}
$e_{1,1} \ldots e_{1,n}$	$e_{1,1}, e_{2,2}, \ldots, e_{n,n}, e_{1,2}, e_{2,3}, \ldots, e_{n-1,n}, e_{1,3}, e_{2,4}, \ldots, e_{n-2,n}, \ldots, e_{1,n-1}, e_{2,n}, e_{1,n}$: a sequence of entries in the event table, as shown in Fig. 4
$e_{x,y}$	An entry representing the trigger-argument relationship between the x-th word w_x and the y-th word w_y in the sentence S^{NE}
$e_{history}$	A sequence of the previously generated entries according to the chain rule
w_{1n}	$w_1, w_2, \ldots, w_n$: a sequence of words in the sentence S^{NE} where n indicates the number of words
w_x	The x-th word in the sentence S^{NE} containing the word itself, its stem, its part-of-speech tag, its form such as capitalization, its named entity tag, and its dependency label
w_{x-i}	The $x-i$-th word on the left context of the word w_x in the sentence S^{NE}
w_{x+i}	The $x+i$-th word on the right context of the word w_x in the sentence S^{NE}
w_{hx}	The head word of the word w_x in the dependency tree generated by a parser
w_{hhx}	The head word of the head word of the word w_x in the dependency tree
w_{dx}	The dependent word of the word w_x in the dependency tree
w_{ddx}	The dependent word of the dependent word of the word w_x in the dependency tree
$w_{x+1\ y-1}$	The inner context between the word w_x and the word w_y in the sentence S^{NE}
$INNER_{dep}(w_x, w_y)$	The inner context between the word w_x and the word w_y in the dependency tree

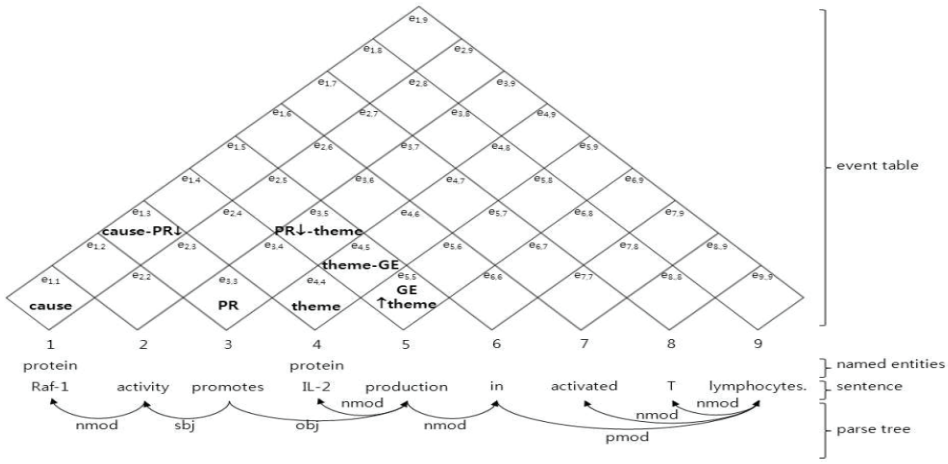


Fig. 4. Event table corresponding to the event set represented in Fig. 1.

As shown in Fig. 4, the event table consists of entries containing the trigger or argument type. Specifically, entry $e_{x,y}$ represents the trigger-argument relationship between the x-th word w_x and the y-th word w_y in the sentence. For example, the entry, $e_{4,4}$ includes the argument type *theme*, which describes that the fourth word *IL*-2 will be used for the theme argument of an event. Also, the entry $e_{5,5}$ contains the type *GE*↑*theme*, which represents that the fifth word *production* will trigger a gene expression event, and then that this gene expression event will be used for the theme argument of another event. In addition, the entry $e_{4,5}$ includes the type *theme-GE*, which describes that the word *production* triggers a complete gene expression event with the *theme* argument *IL*-2. Since the symbol '↓' indicates that one event is divided into more than two binary events, the entry $e_{1,3}$ with *cause-PR*↓ describes that the entry will be combined with other entries, including the symbol '↓'. A more detailed explanation about the event table will be provided in Section 3.3.

3.2 The Maximum Entropy-Based Bio-Molecular Event Extraction Model

In order to solve the event extraction problem by effectively estimating each probabilistic term of Eq. (8), the proposed model utilizes the two words (such as w_x and w_y); the sentence contexts (such as w_{x-2}, w_{x-1}, w_{x+1}, w_{x+2}, w_{y-2}, w_{y-1}, w_{y+1}, w_{y+2}, and w_{x+1y-1}); the dependency contexts (such as w_{hhx}, w_{hx}, w_{dx}, w_{ddx}, w_{hhy}, w_{hy}, w_{dy}, w_{ddy}, and $INNER_{dep}(w_x, w_y)$); and the entry histories (such as $e_{history}$), as represented in the equation below.

$$P\left(e_{x,y} \mid S^{NE}, e_{history}\right)$$

$$= P\left(e_{x,y} \mid w_1, w_2 \ldots, w_n, e_{history}\right) \tag{9}$$

$$\approx P\left(e_{x,y} \left| \begin{matrix} w_x, w_{x-2}, w_{x-1}, w_{x+1}, w_{x+2}, w_{hh_x}, w_{h_x}, w_{d_x}, w_{dd_x}, \\ w_y, w_{y-2}, w_{y-1}, w_{y+1}, w_{y+2}, w_{hh_y}, w_{h_y}, w_{d_y}, w_{dd_y}, \\ e_{history}, w_{x+1y-1}, INNER_{dep}(w_x, w_y) \end{matrix} \right.\right) \tag{10}$$

$$P\left(e_{1,3}\left|\begin{matrix}w_1, \phi, \phi, w_2, w_3, w_3, w_2, \phi, \phi, \\ w_3, w_1, w_2, w_4, w_5, \phi, \phi, w_2, w_1, \\ e_{1,1}, e_{3,3}, w_2, w_2\end{matrix}\right.\right) \tag{11}$$

$$= P\left(cause - PR \downarrow \left|\begin{matrix}Raf - 1, \phi\, \phi\, activity, promotes, promotes, activity, \phi\, \phi \\ promotes, Raf - 1, activity, IL - 2, production, \phi\, \phi\, activity, Raf - 1, \\ cause, PR, activity, activity\end{matrix}\right.\right) \tag{12}$$

Both Eqs. (11) and (12) provide an example, coupled with a detailed explanation. The word w_x includes all of the word itself, its stem, its part-of-speech tag, its form, its named entity tag, and its dependency label, in order to adequately describe the information of the word. The word *Raf*-1 is represented as *Raf*-1, *Raf*-1, *noun*, *Capital*: *Number*, which indicates that the word includes the capital letter *R*, the number "1," *protein*, and *noun*.

Also, the sentence context consists of the left and right hand side words of each word, and the inner words between the word w_x and the word w_y in the sentence. Specifically, the context word w_i indicates nothing ϕ if $0 \geq i$ or $i \geq n$, where n indicates the number of all words in the sentence. Furthermore, the inner words w_{x+1y-1} takes nothing ϕ if $x + 1 > y - 1$. As shown in Eq. (11), the entry $e_{1,3}$ can finally utilize w_1,w_2,w_3,w_4,w_5 as the sentence context.

Additionally, the dependency context is composed of the head and dependent words of each word, and the inner words between the word w_x and the word w_y in the dependency tree. As presented in the sentence context, w_i indicates nothing ϕ if $0 \geq i$, or $i \geq n$ where i substitutes for each of hh_x, h_x, d_x and dd_x. As described in Eq. (11), the entry $e_{1,3}$ can use w_1,w_2,w_3 on the dependency path between the word w_1 and the word w_3.

Finally, the history context represents some useful entries selected from the event entries previously generated by the chain rule. For example, the entry $e_{1,3}$ can utilize the immediate event entries $e_{1,1}$ and $e_{3,3}$.

$$P_{ME}(y|x) = \frac{1}{Z(x)} exp\left(\prod_{i=1}^{k} \lambda_i f_i(x, y)\right) \tag{13}$$

In order to practically calculate Eq. (10), the proposed model adopts the maximum entropy framework [22-25], which is one of the most powerful principles of statistical inference. In the maximum entropy framework, the conditional probability of predicting an outcome y given history x is defined as in Eq. (13). In the equation, $f_i(x, y)$ is the feature function, and λ_i is the weighting parameter of $f_i(x, y)$. Also, k is the number of features, and $Z(x)$ is the normalization factor for $\Pi_y p(x|y)=1$. The maximum entropy framework can select a unique joint probability distribution from the set of all joint probability distributions within a reasonable training time [23]. Also, the framework can use arbitrary feature functions in order to reflect the characteristics of the target domain [24]. The ability of freely choosing feature functions gives maximum entropy the obvious advantage over other machine learning methods.

3.3 Relationship between the Event Table and Events

In this section, we first describe the representation of typical events in the event table. And then, we present how the event table can represent the sentences with a distinctive event structure. Every event consists of a trigger and its arguments, where the trigger always indicates a word, while the argument indicates either a word or other event, as shown in Fig. 5(a). Therefore, an event can be represented as a nonterminal node with a few pointers that indicate a trigger or its argument. Also, a trigger and its argument proteins can be represented as terminal nodes without a pointer. Moreover, each event type can be assigned into the event nonterminal node, while either a trigger type or its argument type can be assigned into the terminal node, as is described in Fig. 5(b). In order to reduce the complexity of solving an event extraction problem by focusing on only the event relationship between an event trigger and each of its arguments, the nonterminal is restricted to having two pointers as presented in Fig. 5(c). These terminal and nonterminal nodes can be assigned into the event table (such as the entry $e_{x,x}$ for a terminal node and the entry $e_{x,y}$ with $x \neq y$ for a nonterminal node). Ultimately, a terminal node for a protein has an argument type, while a terminal node for a trigger has a trigger type. Also, the entry $e_{x,y}$ represents the trigger-argument relationship between the word w_x and the word w_y in the sentence.

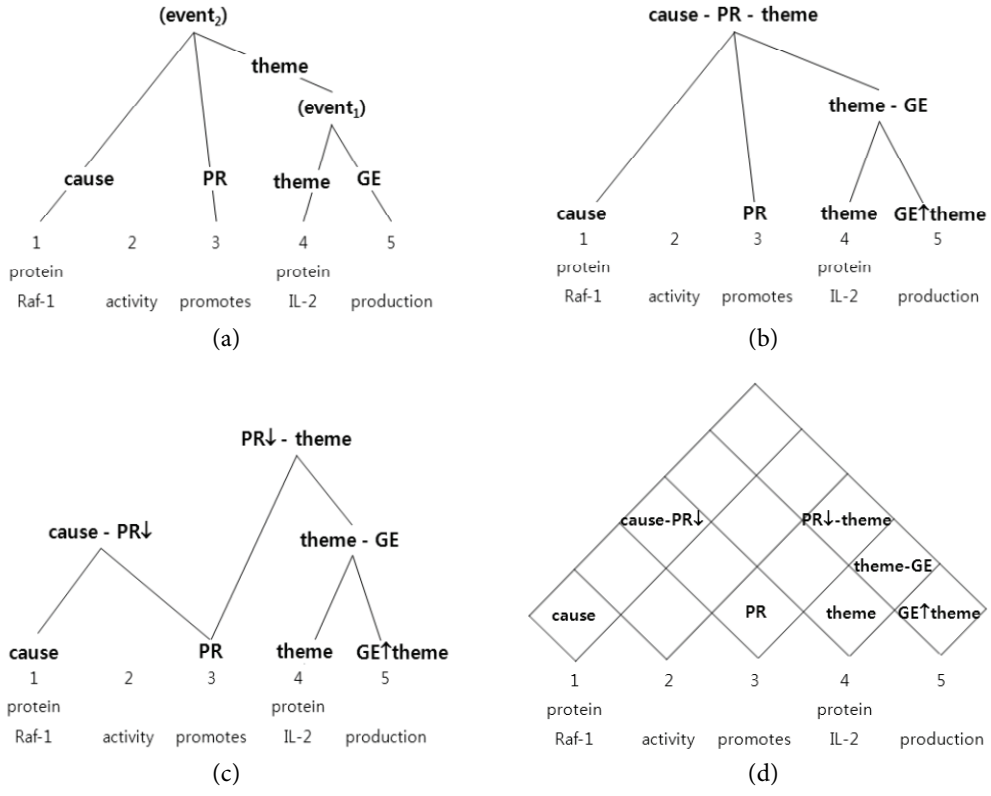
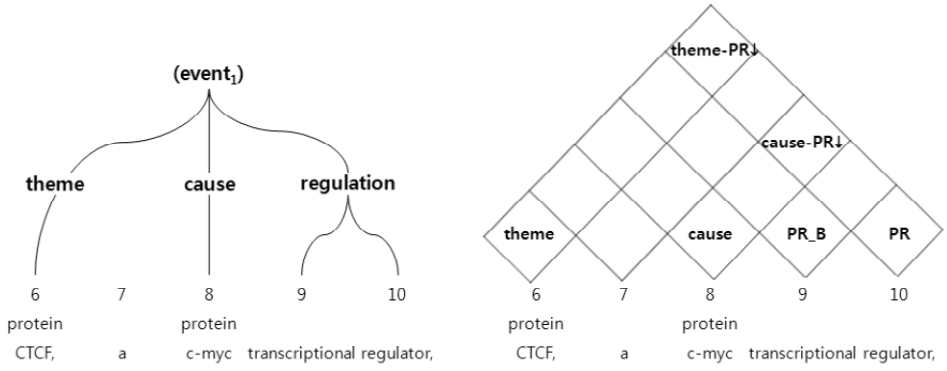
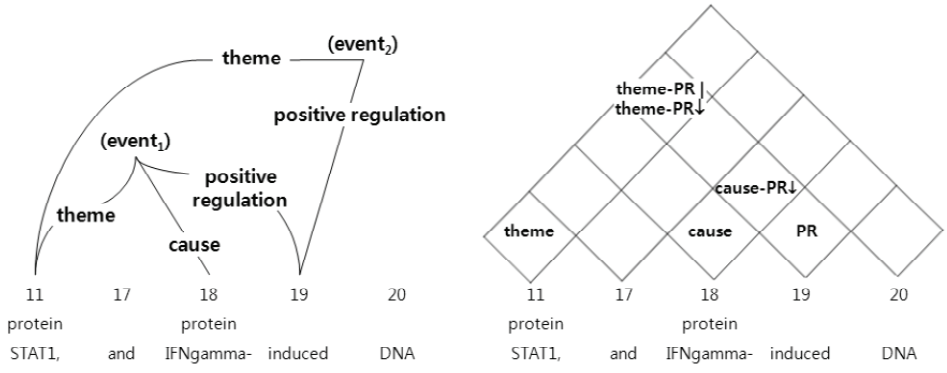


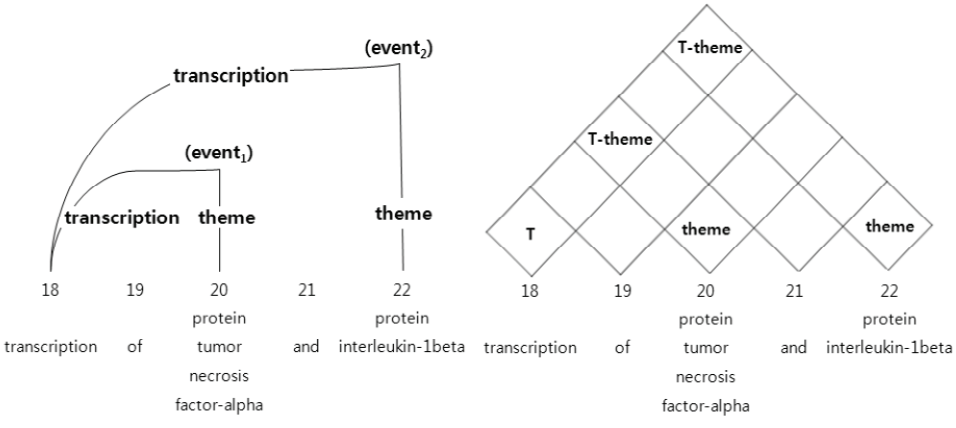
Fig. 5. Process of converting the event set in Fig. 1 into an event table where only the first five words are included in order to compactly represent the events by eliminating numerous entries that are unrelated to the events.



(a)



(b)



(c)

Fig. 6. Event tables that represent the following sentences with distinctive event structures. (a) Differential expression and phosphorylation of CTCF, a c-myc transcriptional regulator, during differentiation of human myeloid cells. (b) This was accomplished by preventing the IFN-induced tyrosine phosphorylation of STAT1, a component of both IFNα- and IFNγ-induced DNA binding complexes. (c) Previously we reported that 3-deazaadenosine (DZA), a potent inhibitor and substrate for S adenosylhomocysteine hydrolase inhibits bacterial lipopolysaccharide-induced transcription of tumor necrosis factor-alpha and interleukin-1beta in mouse macrophage RAW 264.7 cells.

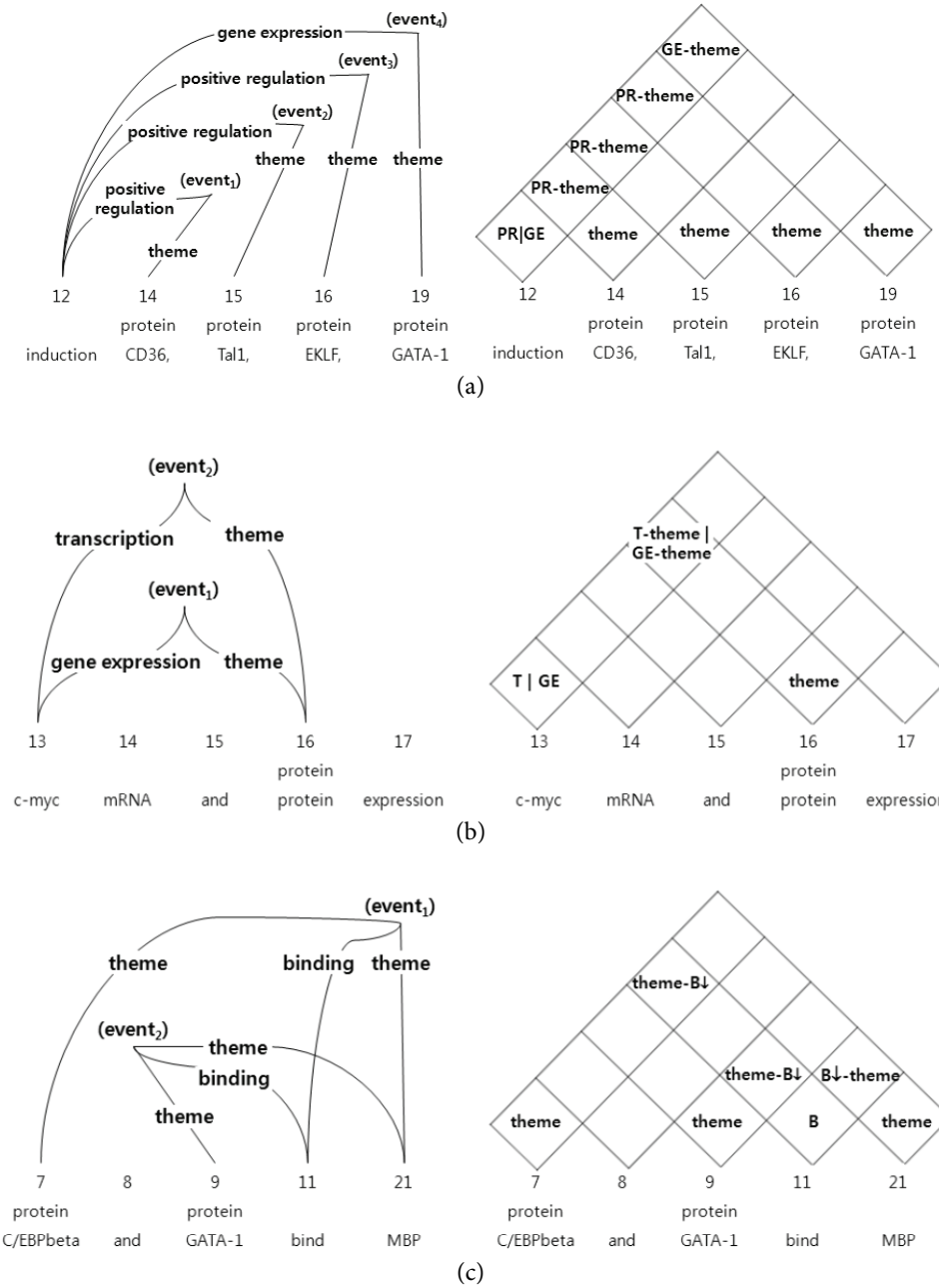


Fig. 7. Event tables that represent the following sentences with distinctive event structures. (a) In early to intermediate stages of erythroid differentiation we monitored the induction of CD36, Tal1, EKLF, NF-E2, and GATA-1 that preceeded expression of EpoR. (b) We show that in the human T lymphoblastic tumor cell line Molt4 c-myc mRNA and protein expression is down-regulated after exposure to dimethyl sulfoxide, to phorbol myristate acetate, or to the calcium ionophore A23187, which raises the intracellular calcium concentration. (c) Furthermore, we have demonstrated that both C/EBPbeta and GATA-1 can bind simultaneously to the C/EBP- and GATA-binding sites in the MBP promoter.

Conversely, three binary events, such as Fig. 5(c), are extracted from the given event table of Fig. 5(d). Then the desired events, such as Fig. 5(a), are obtained by combining two binary events with the symbol '↓' into a single event (such as indicated by the cause- *PR-theme* in Fig. 5(b)). Specifically, every trigger entry includes as much information as possible, because every trigger leads its event. For example, the trigger type *GE*↑*theme* at the event entry $e_{5,5}$ in the event table Fig. 5(d) denotes that the trigger *production* will trigger a gene expression event, and that this gene expression event will be used for the theme argument of another event.

Additionally, the proposed event table can cover some unusual examples. As shown in the multi-word trigger example of Fig. 6(a), the normal trigger type is assigned to the last trigger word, while the trigger type with *B* (Begin) or *I* (Inside) is assigned to other trigger words (shown at the entry $e_{9,9}$). This is because we assumed that the last word of a trigger leads the trigger. On the other hand, a protein has higher priority than a word in the event table, because a word (such as *IFNgamma-induced* in Fig. 6(b)) consists of a protein (such as *IFNgamma*), and a trigger (such as *induced*). The given correct protein consisting of more than two words (such as the entry $e_{20,20}$ at Fig. 6(c)) is handled as one element, in order to reduce the event table size without any event extraction performance loss.

As described in the entry $e_{12,12}$ in Fig. 7(a) and the entry $e_{13,16}$ in Fig. 7(b), the event table utilizes the symbol "|" to represent more than two triggers or event types [8]. In particular, the entry $e_{13,13}$ with *T* |*GE* indicates that the trigger *c-myc* can become a transcription event trigger as well as a gene expression event trigger. Also, the entry $e_{13,16}$ with *T-theme* and *GE-theme* indicates that there is both a transcription event and a gene expression event between the same trigger *c-myc* and the same theme argument *protein*.

The event table allows for element sharing. The shared element can be a single trigger, such as Fig. 6(c) or Fig. 7(a). Although the trigger *induction* of Fig. 7(a) is shared by four events, one leads to a gene expression, while the other three lead to positive regulation events. Though both ($event_1$) and ($event_2$) at Fig. 7(b) take the same trigger and the same argument, they can take the different event types, such as $e_{13,16}$. Thus, part of an event can be shared between two events, such as Fig. 7(c) that ($event_1$) and ($event_2$) take the different theme argument with each other, while they take both the same trigger and the same theme argument.

4. Experiments

In order to examine the practical feasibility of our proposed bio-molecular event extraction model, we evaluated the coverage of the event table and the event extraction performance according to the feature combination. In order to fairly evaluate the proposed model, we utilized the training set, the test set, and the evaluation metrics such as *precision*, *recall*, and *f-score* provided by the BioNLP'09 shared task on event extraction [7].

4.1 Coverage Analysis

For the purpose of examining the coverage of the proposed model, we have applied the correct 8,597 events in the training set to Eqs. (1), (5), and (7), as shown in Table 2, where *Num* indicates the number

of events belonging to each event type. Since Eq. (1) describes the definition of the bio-molecular event extraction problem, there is no coverage loss. Because Eq. (5) cannot extract every event placed in more than two sentences, the coverage of Eq. (5) decreases by 7.99%. Clearly, every trigger and its argument proteins in 529 (6.15%) events are located in different sentences, and 158 (1.84%) events take one of these 529 events as an argument.

Table 2. Coverage per each equation in the training set

Event type	Num	Eq. (1)	Eq. (5)	Eq. (7)
Gene expression	1,738	100.00	94.59	94.19
Transcription	576	100.00	94.79	94.44
Protein catabolism	110	100.00	97.27	96.36
Phosphorylation	165	100.00	95.76	95.15
Localization	263	100.00	96.20	96.20
Binding	880	100.00	94.55	93.41
Regulation	960	100.00	89.48	89.27
Positive regulation	2,843	100.00	91.52	90.75
Negative regulation	1,062	100.00	85.59	84.56
Total	8,597	100.00	92.01	91.36

Moreover, the coverage of Eq. (7) decreases by an additional 0.65%, because the correct event table does not correspond to the set of the correct events, even though the event table can handle some unusual examples, as previously presented in Section 3.3. For example, the event table cannot include the trigger 'expression' in the word 'overexpression' without any protein, because the event table is based on a protein unit or a word unit.

4.2 The Effectiveness of Feature Combination

For the purpose of evaluating the bio-molecular event extraction performance according to the feature combination, we utilized some useful features selected from Eq. (10). We also evaluated the proposed model with these features using 10-fold cross validation on the training set. As described in Table 3, the *word* features indicate w_x and w_y, which are relative to the type of the entry $e_{x,y}$ in the event table. This feature includes the word itself, its stem, its part-of-speech tag, its form, its named entity tag, and its dependency label, as described in Table 1. Then, the *sentence* features describe the sentence context of these two words, while the *dep*endency features represent their dependency context. The *history* features represent some useful entries previously generated by the chain rule.

Table 4 presents a report on the performances of the proposed model on various feature combinations. By adding *sen*tence features or *dep*endency features to *word* features, the performances tend to increase the recall. Especially, the model adding the *sen*tence features improves the recall by approximately 9% on *Regulation* events. These results show that the simple *word* features tend to determine that a given word is a non-trigger word. This is because a trigger word in a sentence frequently occurs as a non-trigger word in other sentences in the training set [8]. However, the model utilizing the *sen*tence features or *dep*endency features comparatively prefer a trigger to an ordinary word based on more precise context information. As the number of correct simple (and binding) events increases, the number of correct regulation events significantly increases by taking these correct simple (and binding) events as arguments. On the other hand, it is remarkable that the sentence features are

more useful than the dependency features, because the dependency features can be related to some errors generated by a dependency parser, while the sentence features are free from these errors.

Table 3. Feature set used in experiments

Category	Practical features	Description
Word features	w_x, w_y	The x-th word and the y-th word in a sentence
	$w_x w_y$	A concatenation of w_x and w_y
Sentence features	w_{x-2}, w_{x-1}	The (x–2)-th word and the (x–1)-th word in a sentence
	w_{x+1}, w_{x+2}	The (x+1)-th word and the (x+2) -th word in a sentence
	$w_{x-1} w_x$	A concatenation of w_{x-1} and w_x
	w_{x-2} w_{x-1} w_x	A concatenation of w_{x-2}, w_{x-1} and w_x
	w_{y-2}, w_{y-1}, w_{y+1}, w_{y+2}	The features obtained by replacing x to y in the ones above
	$w_{y-1} w$	
	$w_{y-2} w_{y-1} w_y$	
	$w_{x+1} w_{x+2}$ ... $w_{y-2} w_{y-1}$	A concatenation of all words from w_{x+1} to w_{y-1}
Dependency features	length of path	The length of the path from w_x to w_y on the dependency tree
History features	$e_{x,x}$, $e_{y,y}$	An entry corresponding to the word w_x and an entry corresponding to the word w_y

Table 4. Performance according to various feature combinations (recall/precision/f-score)

	Simple event	Binding	Regulations	All
word	70.95 / 38.94 / 50.29	34.66 / 18.28 / 23.94	25.89 / 13.19 / 17.48	41.64 / 21.81 / 28.63
word+sen	73.20 / 42.02 / 53.39	39.45 / 18.77 / 25.44	34.67 / 13.98 / 19.92	47.84 / 21.78 / 29.93
word+dep	72.12 / 38.23 / 49.97	38.56 / 17.49 / 24.07	31.55 / 13.17 / 18.58	45.64 / 20.67 / 28.45
word+his	69.89 / 41.37 / 51.98	36.52 / 20.16 / 25.98	28.10 / 12.84 / 17.62	42.74 / 21.55 / 28.65
word+sen+dep	72.85 / 44.47 / 55.23	39.89 / 20.50 / 27.08	33.73 / 15.49 / 21.23	47.25 / 23.92 / 31.76
word+sen+his	71.83 / 48.49 / 57.89	38.83 / 23.57 / 29.34	34.65 / 14.94 / 20.88	47.31 / 24.00 / 31.85
word+dep+his	69.77 / 42.55 / 52.87	36.35 / 20.81 / 26.47	28.70 / 14.64 / 19.39	43.02 / 23.49 / 30.39
word+sen+dep+his	72.00 / 48.99 / 58.31	39.54 / 23.71 / 29.64	35.30 / 15.18 / 21.23	47.81 / 24.24 / 32.17

As compared with the model using *word* features and *sen*tence features, the model adding *dep*endency features or *history* features improves the precision because of the following reasons since the distance between a trigger and its argument is much closer on the dependency tree than on the raw sentence [8], the model can actually focus on candidate events on the short dependency distance by utilizing the *dep*endency features. In addition, the model can more accurately find the event type by using *history* features since the *history* features emphasize the trigger type and the argument type, both of which compose the event type. Finally, Table 4 shows that the model using all kinds of features performs best at a 32.17% *f-score* because more features generally lead to better performance.

For comparison with previous bio-molecular event extraction models in the same test environment, we evaluated the proposed model on the official evaluation metrics provided by the BioNLP'09 shared task [7]. The proposed model achieved a 50.44% recall, 26.69% precision, and 34.91% *f*-score in the approximate recursive matching. Compared to the results [7] of other BioNLP'09 shared task participants, the proposed model ranks in the upper-middle range.

5. Conclusion

In this paper, we proposed a maximum entropy-based model for bio-molecular event extraction. The proposed event extraction model has the desirable characteristics, which are described below.

First, we can clearly describe the bio-molecular event extraction problem by mathematical modeling. We mathematically defined the problem to extract the bio-molecular events from a document. We also described a method of deriving the proposed model, which can utilize many useful features, such as sentence contexts and dependency contexts, from the definition.

Second, we can claim that it is possible to replace the bio-molecular event extraction problem with the event table generation problem. In the proposed event table, each unary entry is assigned to each word in the sentence, and each binary entry represents the trigger-argument relationship between the given pair of words. The event table can cover some unusual events, such as the multi-word trigger or element sharing, as described in Figs. 6 and 7. Our experimental results showed that the coverage of bio-molecular event extraction decreases only by 0.65%. This is because the correct event table does not correspond to the set of the correct events.

Third, we can simplify the process of solving a bio-molecular event extraction problem because the proposed model focuses only on the individual binary relationship between an event trigger and each of its arguments. Specifically, we replaced the event extraction problem with the event table generation problem, and then we decomposed the event table generation problem into the small manageable problems of filling each entry into the event table.

Fourth, we verified the useful feature combination by analyzing the event extraction power of each feature combination. We have found that more features mainly tend to significantly increasing the precision of the bio-molecular event extraction problem, even though more features slightly decreased the recall in some cases. Furthermore, it is remarkable that the sentence features are more useful than the dependency features, since the dependency features can be related to some errors generated by a dependency parser, while the sentence features are free from these errors.

Unlike previous approaches that have intuitively designed a pipeline consisting of a trigger detection step and an argument recognition step, the proposed model combines these two steps by using the event table. Also, the model is designed based on the mathematical derivation process rather than on intuition. Furthermore, the proposed model can describe why these two steps can utilize the same machine learning techniques and similar features.

For future research, we would like to study a method of selecting more appropriate features for the maximum entropy model, in order to extract the events more precisely. In addition, we aim at applying an anaphora resolution method to the bio-molecular event extraction problem for covering some events in which a trigger and arguments are located in different sentences.

References

[1] C. Blaschke, M.A. Andrade, C. Ouzounis, and A. Valencia, "Automatic extraction of biological information from scientific text: protein-protein interactions," in *Proceedings of International Conference on Intelligent Systems for Molecular Biology*, Germany, 1999, pp. 60–67.

[2] R. Bunescu, R. Mooney, A. Ramani, and E. Marcotte, "Integrating co-occurrence statistics with information

extraction for robust retrieval of protein interactions from medline," in *Proceedings of the Workshop on Linking Natural Language Processing and Biology*, New York, NY, 2006, pp. 49–56.

[3] H. W. Chun, Y. Tsuruoka, J.D. Kim, R. Shiba, N. Nagata, T. Hishiki, and J. Tsujii, "Extraction of gene-disease relations from medline using domain dictionaries and machine learning," in *Proceedings of the Pacific Symposium on Biocomputing,* Maui, HI, 2006, pp. 4–15.

[4] F. Rinaldi, G. Schneider, K. Kaljurand, M. Hess, C. Andronis, O. Konstandi, and A. Persidis, "Mining of relations between proteins over biomedical scientific literature using a deep-linguistic approach," *Artificial Intelligence in Medicine* vol. 39, no. 2, 2007, pp. 127–136.

[5] A. Airola, S. Pyysalo, J. Bjorne, T. Pahikkala, F. Ginter, and T. Salakoski, "All-Paths Graph Kernel for Protein-Protein Interaction Extraction with Evaluation of Cross-corpus Learning," *BMC Bioinformatics* vol. 9, Suppl. 11, 2008.

[6] K. M. Park, H. C. Cho, and H. C. Rim, "Utilizing various natural language processing techniques for biomedical interaction extraction," *Journal of Information Processing Systems,* vol. 7, no. 3, pp. 459–472, 2011.

[7] J. D. Kim, T. Ohta, S. Pyysalo, Y. Kano, and J. Tsujii, "Extracting bio-molecular events from literature - the Bionlp'09 Shared Task," *Computational Intelligence*, vol. 27, no. 4, pp. 513–540, 2011.

[8] J. Björne, J. Heimonen, F. Ginter, A. Airola, T. Pahikkala, and T. Salakoski, "Extracting contextualized complex biological events with rich graph-based feature sets," *Computational Intelligence*, vol. 27, no. 4, pp. 541–557, 2011.

[9] E. Buyko, E. Faessler, J. Wermter, and U. Hahn, "Syntactic simplification and semantic enrichment - trimming dependency graphs for event extraction," *Computational Intelligence*, vol. 27, no. 4, pp. 610–644, 2011.

[10] K. Hacioglu, "Semantic Role Labeling Using Dependency Trees," in *Proceedings of the International Conference on Computational Linguistics*, Geneva, Switzerland, 2004.

[11] R. T. H. Tsai, W. C. Chou, Y. C. Lin, C. L. Sung, W. Ku, Y. S. Su, T. Y. Sung, and W. L. Hsu, "Biosmile: adapting semantic role labeling for biomedical verbs: an exponential model coupled with automatically generated template features," in *Proceedings of the HLT-NAACL BioNLP Workshop on Linking Natural Language and Biology,* New York, NY, 2006, pp. 57–64.

[12] K. Fundel, R. Kuffner, and R. Zimmer, "RelEx-relation extraction using dependency parse trees," *Bioinformatics,* vol. 23, no. 3, pp. 365–371, 2007.

[13] H. Kilicoglu and S. Bergler, "Effective bio-event extraction using trigger words and syntactic dependencies," *Computational Intelligence*, vol. 27, no. 4, pp. 583–609, 2011.

[14] J. Hakenberg, I. Solt, D. Tikk, V. H. Nguyên, L. Tari, Q. L. Nguyen, C. Baral, and U. Leser, "Molecular event extraction from link grammar parse trees in the BioNLP'09 Shared Task," *Computational Intelligence*, vol. 27, no. 4, pp. 665–680, 2011.

[15] A. Vlachos, P. Buttery, D. O. Séaghdha, and T. Briscoe, "Biomedical event extraction without training data," in *Proceedings of the BioNLP 2009 Workshop Companion Volume for Shared Task,* Boulder, Colorado, 2009, pp. 37–40.

[16] R. Morante, V. Van Asch, and W. Daelemans," A memory-based learning approach to event extraction in biomedical texts," in *Proceedings of the BioNLP 2009Workshop Companion Volume for Shared Task,* Boulder, Colorado, 2009, pp. 59–67.

[17] H. G. Lee, H. C. Cho, M. J. Kim, J. Y. Lee, G. Hong, and H. C. Rim, "A multi-phase approach to biomedical event extraction," in *Proceedings of the BioNLP 2009Workshop Companion Volume for Shared Task,* Boulder, Colorado, 2009, pp. 107–110.

[18] S. Riedel, R. Sætre, H. W. Chun, T. Takagi, and J. Tsujii, "Bio-molecular event extraction with Markov logic," *Computational Intelligence*, vol. 27, no. 4, pp. 558–582, 2011.

[19] S. Van Landeghem, B. De Baets, Y. Van de Peer, and Y. Saeys, "High-precision bio-molecular event extraction from text using parallel binary classifiers," *Computational Intelligence*, vol. 27, no. 4, pp. 645–664, 2011.

[20] K. Sagae and A. Lavie, "A best-first probabilistic shift-reduce parser," in *Proceedings of the COLING/ACL on Main Conference Poster Sessions,* Sydney, Austrailia, 2006, pp. 691–698.

[21] E. Charniak, S. Goldwater, and M. Johnson, "Edgebased best-first chart parsing," in *Proceedings of the Sixth Workshop on Very Large Corpora,* Montreal, Canada, 1998, pp. 127–133.

[22] A. L. Berger, V. J. D. Pietra, and S. A. D. Pietra, "A maximum entropy approach to natural language processing," *Computational Linguistics* vol. 22, no.1, pp. 39–71, 1996.

[23] T. Lukasiewicz, "Credal networks under maximum entropy," in *Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence*, San Francisco, CA, 2000, pp. 363–370.

[24] L. Zhang, *Maximum Entropy Modeling Toolkit for Python and C++.* Shenyang: Natural Language Processing Lab., Northeastern University, 2004.

[25] A. Ratnaparkhi, "Learning to parse natural language with maximum entropy models," *Machine Learning,* vol. 34, no. 1-3, pp. 151–175, 1999.

Hyoung-Gyu Lee

He graduated from Korea University with a B.S. degree in Computer Science. He is currently Ph.D. candidate of Department of Computer and Radio Communications Engineering in Graduate School at Korea University. He is also a member of Natural Language Processing Laboratory at Korea University. His research interests are in areas of natural language processing, including statistical machine translation, biomedical text mining, information extraction, and information retrieval.

So-Young Park http://orcid.org/0000-0003-0746-218X

She received her B.S. degree in computer science and engineering from SangMyung University in 1997, and both M.S. and Ph.D. degrees in computer science from Korea University in 1999 and 2005, respectively. She has been an associate professor in the department of game design & development at SangMyung University since 2007. Her current research interests include natural language understanding and data mining.

Hae-Chang Rim

He has received his Ph.D. in Computer Science from the University of Texas at Austin in 1990. He is currently Professor of Computer Science in the College of Information and Communications at Korea University and serves as Research Director of Natural Language Processing Laboratory at Korea University. His primary research interests lie in several areas of computational linguistics, including natural language processing, Korean language processing, and information retrieval.

Do-Gil Lee

He received his M.S. and Ph.D. degrees in computer science from Korea University, in 2001 and 2005, respectively. He was a research engineer at the NHN Corporation. He is an HK professor at Research Institute of Korean Studies, Korea University. His research interests include developing probabilistic methods in Korean language processing.

Hong-Woo Chun

He received the B.S. and M.Sc. degrees in Computer Science & Engineering from Korea University and the Ph.D. in Graduate School of Information Science and Technology from The University of Tokyo in 2002, 2004, and 2007, respectively. He worked for JBIRC, AIST as a postdoctoral researcher from 2007 to 2008, and worked for DBCLS and ROIS as a project researcher from 2008 to 2010. He is currently working for KISTI as a senior researcher. His research interests are Natural Language Processing, Machine Learning, and Text Mining.