Biomedical Information Extraction with Predicate-Argument Structure Patterns

Akane Yakushiji [a] **Yusuke Miyao** [a] **Yuka Tateisi** [b,a] **Jun'ichi Tsujii** [a,b]

[a] Department of Computer Science, University of Tokyo, [b] CREST, JST (Japan Science and Technology Agency)

Abstract

Due to the ever growing amount of publications, Information Extraction (IE) from text is increasingly is recognized as one of crucial technologies in bioinformatics. However, for IE to be practically applicable, adaptability/portability of a system is crucial, considering extremely diverse demands in biomedical IE application. We should be able to construct a set of "extraction rules" adapted for a specific application at low cost.

We propose a new method for automatic construction of application-specific extraction rules, which effectively utilizes predicate-argument structures (PASs) produced by a full-parser. By dividing labor between generic linguistic rules in the parser and application-specific extraction rules to be constructed from scratch, this method facilitates acquisition of extraction rules from a relatively small annotated corpus. We conducted an experiment in which the method was applied to extraction of protein-protein interaction. The result shows that, though the current version of the construction algorithm is straightforward, the performance is remarkably promising, comparable with those obtained by manual-made extraction rules or those obtained by rules generalized by machine learning techniques.

Introduction

Although Information Extraction (IE) from text is increasingly recognized as a crucial component in bioinformatics, it has hardly been used yet in the process of actual data curation, knowledge integration/discovery, etc. This is because

(1) [Quality] the performance of current IE in terms of recall and precision is not good enough, and/or

(2) [Portability/Adaptability] current IE requires a lot of human effort to adapt for particular information needs in specific applications.

While IE systems whose *extraction rules* are carefully crafted and adapted for particular applications (such as [8, 3]) show near-practical performance, the same performance can hardly be repeated in different applications that have to deal with different kinds of information (protein-protein interaction, disease-gene association, toxicity of materials, etc.). Manual engineering of IE systems is a tedious and time-consuming process.

Techniques based on machine learning (ML) (such as [6, 2]) are expected to alleviate this difficulty in manually-crafted IE. However, in most cases, they simply transfer the cost of manually crafting rules to that of constructing a large amount of training data, which in case of IE requires tedious manual labor of annotating text. It is also the case that, when they are applied directly to surface sequences of words in text, ML techniques as they are have shown poor results.

In order to render IE techniques practical in biomedical domains, it is crucial that a generic part of a system, which can be transferred across IE systems in different applications, is clearly distinguished from application-specific part and thereby the cost of adaptation could be minimized.

In this paper, we propose a new system architecture, in which a full parser plays a significant role for improving the quality of performance as well as increasing the adaptability of an IE system. A full parser is a program that takes a sentence as input to produce its semantic representation (predicate-argument structure: PAS). While a full parser embodies linguistic knowledge that is valid across different applications, extraction rules that are application-dependent have to be constructed from scratch.

Because diverse forms of surface sentences with the same meanings are reduced into single PASs by a full parser, the construction algorithm for extraction rules is much simpler than those seeing sentences as mere sequences of words and can acquire rules by using a much smaller training set. The rules constructed thus give a promising performance (37.3% precision and 45.3% recall without any manual intervention for IE of protein-protein interactions). Furthermore, while we do not discuss in this paper, because extraction rules thus acquired are easy to understand, one can revise and augment them manually and develop IE systems with performance comparable with (or better than) carefully crafted IE systems.

This paper discusses details of the construction algorithms, the performance of an IE system and future de-

velopment after briefly discussing the full parser.

Previous Work

Research for biomedical interaction extraction from text is now attracting many works [4, 13, 12, 1, 14, 9, 8, 17, 3, 2, 6, 20]. Their IE systems include a process that reduces diverse surface forms in text into a standard structure by natural language processing (NLP) and makes extraction rules on the structure. There are works using pattern matching [12, 1, 2, 6] and ones using shallow parsing [14, 9, 8] or full parsing [21, 13, 4, 17, 3]. Another categorization of the works is how they construct extraction rules. One approach is based on hand-written rule sets [12, 4, 1, 14, 17, 8, 3]. The other is rule generation by ML based on a corpus with desired information [9, 2, 6]. Some latest works related closely to ours are as follows.

Daraselia et al. [3] used a full parser based on context-free grammar and a lexicon developed specifically for MEDLINE. They wrote extraction rules on semantic trees and extracted mammalian protein functional links by 91% precision and the estimated recall was 30–50%. Their extraction rules require much manual modification to apply to different kinds of information.

Bunescu et al. [2] used machine learning technique to construct extraction rules on surface words as inter-fillers (text fragments between participating entities), role-fillers and longest common subsequences which represent protein-protein interactions. The corpus they used is Aimed, which consists of 230 MEDLINE abstracts annotated with protein names and protein-protein interactions. They reached about 48% precision for 45% recall. One of the shortcomings of the system is that generated patterns are hard to augment manually to improve performance because the patterns are not ensured syntactically or semantically. Huang et al. [6] used a dynamic programming algorithm to obtain patterns on parts-of-speech and surface words for protein-protein interactions. On sentences which include keywords, their precision was 80.5% and recall was 80.0%. The system requires training corpus on which sentences are aligned to estimate parameters.

Besides the biomedical domain, there are works which acquire extraction rules automatically in other domains. EXDISCO by Yangarber et al. [22] identifies a set of relevant documents and a set of extraction rules from un-annotated text, starting from a small set of seed rules. The rules are constructed on results of a general-purpose dependency parse. Their result was 73% precision and 57% recall on MUC-6 corpus [11]. Sudo et al. [16] acquired extraction rules as subtrees derived from dependency trees of sentences in automatically retrieved un-annotated documents. Their result was about 75% precision and 55% recall on the Management Succession scenario of MUC-6. One problem of those systems is that dependency parse cannot treat non-local dependencies such as the semantic subject (*Entity1*) of "activating" in the last sentence of Table 1, and thus rules acquired from the constructions are partial.

activate
ARG1 *Entity1 (semantical subject)*
ARG2 *Entity2 (semantical object)*

Figure 1: A PAS of "**activate**"

Approach

Surface Variations of the Same Information

IE can be seen as a process that reduces diverse surface forms in text into a fixed standard representation when they express the same information. Whether two forms in text express the same information or not depends on the perspectives or interests researchers have. For example, "*Entity1* interacts with non-polymorphic regions of *Entity2*" can be considered to express the same information as "*Entity1* interacts with *Entity2*" if one is interested in general protein-protein interaction regardless of their modes of interaction, but cannot be for others whose interests are the modes. In short, the application-specific nature of IE resides in this kind of perspective-dependency in the definition of information.

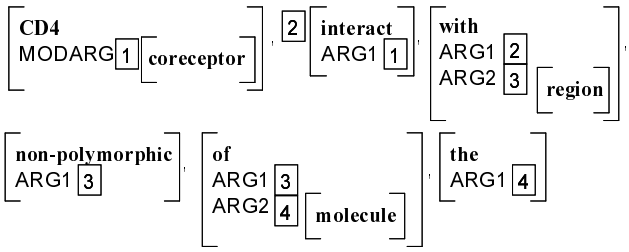
However, there are other types of surface variations that express the same information regardless of users' perspectives, such as "*Entity1* activates *Entity2*" and "*Entity2* is activated by *Entity1*". In some cases, a surface form can be considered to contain as its part the same information that another form expresses, regardless of users' perspectives. "*Entity1* can activate *Entity2*" vs. "*Entity1* activates *Entity2*" are such examples.

We mean that by a full parser, a program which assigns standard forms to surface sentences, the same information of these kinds is represented in the same formats. In this format, all the surface forms in Table 1 share the same information "*Entity1* activate *Entity2*" as their part, and the shared information is represented in the same form in Figure 1. We call this form Predicate-Argument Structure (PAS).

An example of a set of PASs that are assigned to a sentence is given in Figure 2. **Bold words** are predicates. Arguments of the predicates are described in ARGn ($n = 1, 2, \ldots$). MODARG denotes the modified PAS. Numbers in squares denote shared structures, i.e. the same PAS.

Table 1: Syntactical variation examples of "activate"

Active Main Verb	*Entity1* recognizes and **activates** *Entity2*.
After an Auxiliary Verb	*Entity1* can **activate** *Entity2* through a region in its carboxy terminus.
Passive	*Entity2* are **activated** by *Entity1a* and *Entity1b*
Past Particle	*Entity2* **activated** by *Entity1* are not well characterized.
Verb of a Relative Clause	The herpesvirus encodes a functional *Entity1* that **activates** human *Entity2*.
Infinitive	*Entity1* can functionally cooperate to synergistically **activate** *Entity2*.
Gerund in a prepositional phrase	The *Entity1* play key roles by **activating** *Entity2*.



CD4 coreceptor interacts with non-polymorphic regions of the molecules.

Figure 2: PAS example

What is important here is that computation from surface sentences to PASs can be carried out regardless of users' perspectives and that PASs represent by single forms the same information that appear in very different sequences of words. Due to such reduction in complexity, we can expect that the construction algorithm of IE rules that works on PASs needs a much smaller training corpus than those working on surface word sequences. Furthermore, due to the reduction of surface diversity at the PAS level, an IE system with extraction rules at this level should show improved performance in terms of recall.

Properties of Required PAS Patterns

Classification of Required Patterns Although many previous biomedical IE system focus on verbs which represent target events by themselves (i.e. "activate", "bind"), there are many cases that combinations of verbs and certain nouns form proper IE patterns (i.e. "form complexes with", "be considered as an antagonist for"). We investigated and classified patterns which are needed to extract interacting protein pairs occurred in Aimed [2] to see what patterns are required in addition to ones that consist of only one verb. We found five classes based on constituents of the patterns. Table 2 shows the details.

Class (1) consists of the simplest patterns, which include only one verb and interacting proteins (entities) and an optional preposition. The patterns are very classical.

Class (2) includes other patterns with only one verb, and can be divided into two subclasses based on properties of their nominal constituents. The nouns of subclass (2a) cannot be omitted, although ones of subclass (2b) can be omitted and the patterns become class (1). In subclass (2a), a pattern consists of a verb and nouns which are not the entities (here, the interacting proteins) themselves. The nouns tend to represent complexes (ex. "complex"), interacting substances (ex. "receptor", "antagonist") or words denoting interactions (ex. "activity", "interaction"). In subclass (2b), nouns to represent parts or complexes of the entities (ex. "dimer", "region") are inserted.

Class (3) is a complicated case which takes more than one verb in patterns. In most cases, one of the verbs in the patterns is used to modify a noun phrase.

Patterns in class (4) do not include verb phrases and the meaning of them is mainly represented by nouns. Patterns in subclass (4a) consist of nouns representing interacting substances (ex. "ligand", "receptor") and their coordinates. Ones in subclass (4b) are based on nouns representing interaction themselves (ex. "interaction", "oligomerization") or its related things (ex. "complex", "homodimer"). On the other hand, ones in subclass (4c) require other adjective words representing interactions in addition to nouns, although whole patterns are noun phrases.

And patterns in class (5) include adjectives as their key constituents. The adjectives can be both attributive and predicative.

Table 2: Classes of Proper Patterns

(1) *Entity*-Verb(-Preposition)-*Entity*
This study demonstrates that *Entity1* **recognizes** *Entity2a* and *Entity2b* by distinct mechanisms.
We also found another armadillo-protein, *Entity1*, **interacted with** *Entity2*.
(2) Other Patterns with Only 1 Verb
(2a) Combinations of Verbs and Non-Entity Nouns
Cell surface *Entity1* **formed complexes with** actin-binding protein *Entity2*.
Entity1 **was** first **characterized as** a **receptor for** *Entity2*.
Entity1 can **decrease** the fusogenic **activity of** *Entity2* via a direct interaction.
(2b) Nouns Representing Parts etc. of *Entity*
Two *Entity1* **molecules grasp** each **side of** a twofold symmetric *Entity2* **dimer**
Entity1 is **interacted with** a hydrophilic loop **region in** the C-terminal **fragment of** *Entity2*.
(3) Patterns with More than 1 Verbs
Entity1 **recognizes** one FGFR **isoform known as** the *Entity2* isoform.
Entity1 **contains** this primary **site as well as** a **region** that **restricts interaction with** *Entity2*.
(4) Noun Patterns
(4a) Coordinates with Nouns Representing Interacting Substances
Entity1 **ligand (** *Entity2* **)**
cloning of *Entity1* **,** a **ligand for** *Entity2* on human T cells
(4b) Nouns Representing Interaction
Entity1 **-** *Entity2* **complexes**
interaction of *Entity1* **with** *Entity2*
(4c) Nouns and their Modifiers Representing Interaction
Entity1 **and its binding specificity with** *Entity2*
Entity1 **binding domain on** the human *Entity2*
(5) Adjective Patterns
dimeric *Entity1*
Entity1 **is** a **homodimeric** cysteine knot **protein**

Table 3: Additional Features of Proper Patterns

Coordination/Parenthesis
This study demonstrates that *Entity1* **recognizes** *Entity2a* and *Entity2b* by distinct mechanisms.
Anaphora
Entity1 **and its binding specificity with** *Entity2*
Entity in an Optional Prepositional Phrase
Unlike human *Entity1*, the viral cytokine largely uses hydrophobic amino acids to **contact** *Entity2*.

Note that there is a case where only one entity participates in an interaction. An example is the first one of class (5), where only one entity (two molecules of *Entity1*) forms a dimer.

How to Construct Patterns of Each Class Based of these properties of the classes, we can construct patterns of the classes automatically from a corpus tagged with interacting entity information. Main ideas for constructing each class are described below.

Class (1) is simple and easy to extract. Subclasses (4a) and (4b) are depend on certain words, and able to extract only by surface words.

Patterns of subclasses (2a) and (2b) can be divided into components: a verb (and a preposition) (*verbal component*), and noun phrases including entities (*nominal components*). The difference of subclasses (2a) and (2b) is that the nominal components is omissible or not, and subclass (2b) becomes class (1) after omission. In addition, the nominal components of subclass (2b) are more flexible on which verb to connect. Thus the nominal components of subclass (2b) are pure components of the patterns, and ones of subclass (2a) are idiom-like with verbs. Because of this property, we can assume two schemes: Subclasses (2a) and (2b) can be distinguished by whether the nominal components are omissible, and after collecting the nominal components of subclass (2b), they are able to use in combination with any verbs of class (1). At present, we implemented division of patterns of class (2) into verbal and nominal components. But distinction of subclass (2a) from subclass (2b) remains future work.

Although patterns of class (3) are small in number and we have not fully investigated their property, we can observe that they include two kinds of verbs, interaction verbs and general verbs. The interaction verbs are the same with classes (1) and (2b). Thus this class can also be divided into components, although this division is future work.

Classes (4c) and (5) have the same property that adjective modifiers are important. Because most adjectives are general, optional and not appropriate for pattern constituents, we have to filter out such ones. Moreover, we have to find out automatically whether adjectives are needed or more nouns or verbs are needed. This remains future work.

Additional Features Moreover, there are some additional features notable about proper patterns. One is coordination and parenthesis. In addition to class (4a) above, all classes can be combined with coordination or parenthesis. Another feature is anaphora. Patterns including anaphora have a property different from classes in Table 2. The other is optional prepositional phrases, which include one of entities. The entity is somehow the semantic subject of an interaction verb, but it is hard to find out by parsing. Examples of each feature are shown in Table 3. Among the features, we have implemented a module to process coordination and parenthesis (described in the next section). Patterns with the optional prepositional phrases can be extracted in our present method, if relations of main verbs and the prepositional phrase are acquired correctly by parsing. Anaphora handling is still future work.

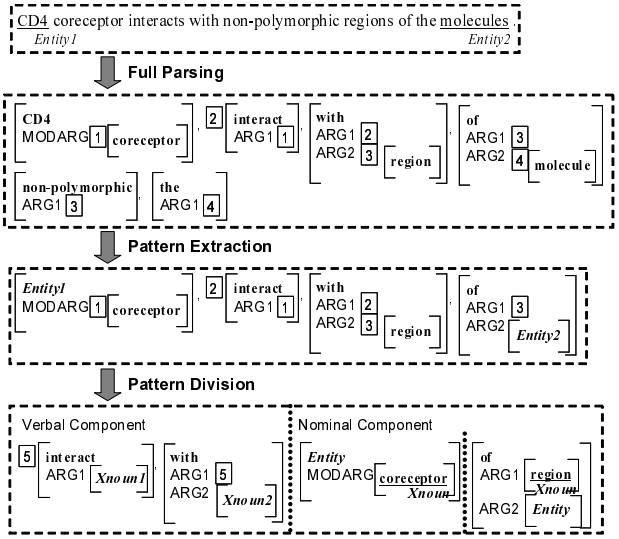


Figure 3: Construction of an Extraction Rule

Method

We automatically construct rules for extraction of protein-protein interactions from an annotated corpus. The corpus needs to be tagged to denote which words represent interacting protein pairs. Sentences of the corpus are passed to a full parser and we convert them into PASs, to absorb syntactical variations. From the PASs, we extract PAS patterns for the interacting proteins and divide them to verbal and nominal components. An example of construction of extraction rules is shown in Figure 3. The details of the construction process are described in the following subsections.

Full Parsing

As we see in the previous sections, a full parser plays a central role in our system. As a parser, we use Enju [18], that is based on Head-Driven Phrase Structure Grammar [15], and is trained on a general English corpus, the Penn Treebank [10]. That is, general linguistic rules in Enju that transforms surface forms to PASs have been acquired from tree banks of articles

in Wall Street Journal (WSJ). Furthermore, the probabilistic model (Maximum Entropy Model) for selecting the most feasible PASs, corresponding interpretations, is also learned by the same corpora. Accompanied with Enju, we use a part-of-speech tagger trained on the GENIA corpus [7] that consists of tagged MEDLINE abstracts.

As expected, the performance of Enju on sentences in WSJ is better than on those in MEDLINE abstracts, but rather surprisingly, the deterioration rate is found to be very small. 83.7% of all PASs in text are correctly recognized for MEDLINE abstracts, tested on the GENIA Treebank [5], while 87.2% for those in WSJ. This fact shows that the linguistic rules embodied in a full parser like Enju, together with probabilistic models learned by ML, are mostly valid across different subject domains and text types.

Pattern Extraction

From PASs which we obtained from a sentence tagged with interacting protein pairs, we extract PAS patterns based on the observed properties in the previous section.

Extraction of Raw PAS Pattern After parsing, we extract the smallest set $(p_0, p_1, \ldots, p_n)$ of PASs which are in *inclusion relation* and includes words that denote interacting proteins, and make it a raw pattern. If an interacting protein is represented in more than one word, we choose the last word as the representative. The process to obtain a raw pattern $(p_0, p_1, \ldots, p_n)$ is as follows.

$including(p)$: PASs which include p as their argument or modify p

$included(p)$: PASs which p includes as its arguments or modifies

1. $p_i = p_0$ is the PAS of a word correspondent to one of interacting proteins, and we obtain candidates of the pattern as the following process:

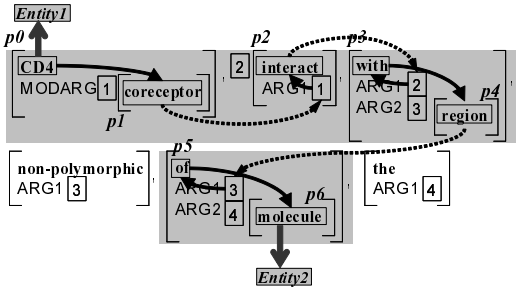
1-1. If p_i is of the word of the other interacting protein, $(p_0, \ldots, p_i)$ is a candidate of the pattern.

1-2. If not, make (a) pattern candidate(s) for each $p_{i+1} \in including(p_i) \cup included(p_i) - \{p_0, \ldots, p_i\}$ by returning to 1-1.

2. Select the smallest pattern candidate as the raw PAS pattern.

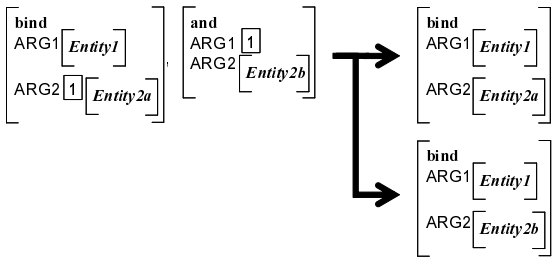
3. Substitute variables (*Entity1*, *Entity2*) for the predicates of PASs correspondent to the interacting proteins.

If an interaction representation includes only one protein (e.g. "*Entity1* dimerization"), we treat the PAS corresponding to the protein word as the raw pattern and obtain the appropriate pattern by the next extending process (See the next subsection).



$\underline{CD4}_{Entity1}$ **coreceptor interacts with** non-polymorphic **regions of** the $\underline{molecules}_{Entity2}$.

Figure 4: Extraction of a PAS Pattern

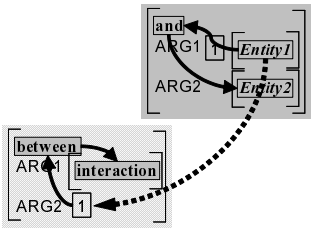


Entity1 bind *Entity2a* and *Entity2b*

Figure 5: Shortcut of a PAS Pattern

Figure 4 shows an example of extraction of a raw PAS pattern. "*CD4*" and "*molecules*" are words representing interacting proteins. First, we set the PAS of "CD4" as p_0. $included(p_0)$ includes the PAS of "coreceptor", and set it as p_1 (shown as a solid arrow). Next, $including(p_1)$ includes the PAS of "interact" (shown as a dotted arrow), so we set it as p_2 (shown as the next solid arrow). We continue similarly until we reach the PAS of "molecules". The result of the extracted raw PAS pattern is shadowed PASs $(p_0, \ldots, p_6)$ with substituting "*CD4*" and "*molecule*" to variables *Entity1* and *Entity2*.

Shortcut and Extension of PAS Patterns As shown in the previous section, representations including appositions by coordinations or parentheses are frequent for protein-protein interactions. We enumerate patterns for all pairs of interacting proteins in such representations by dividing the raw PAS patterns and *shorten* each pattern by substituting PASs of words nearer to the interacting protein for PASs of the appositive words. An example is shown in Figure 5. A pattern for the interacting protein pair (*Entity1*, *Entity2b*) is the lower one, obtained by substituting the PAS of



interaction between *Entity1* and *Entity2*

Figure 6: Extension of a PAS Pattern

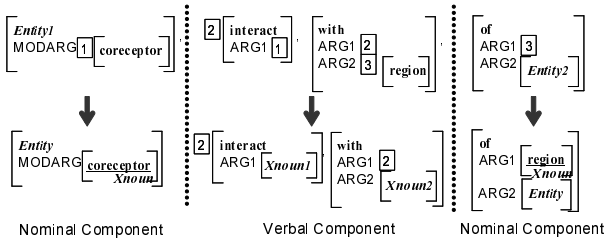


Figure 7: Division of a PAS Pattern

Entity2b for the PAS of *Entity2a* (1) in the PAS of "bind".

Furthermore, there are some cases that interacting proteins are connected directly by a conjunction (e.g. "(interaction between) *Entity1* and *Entity2*"), or only one protein participates in an interaction. These *Entity*-Conjunction-*Entity* patterns and *Entity* patterns cause a lot of errors, thus we extend PAS patterns to ensure they include verbs or nouns other than interacting proteins. We extend a PAS pattern by restarting Step 1 from the head PAS of the raw pattern to a verb or a noun except interacting proteins.

Figure 6 illustrates an example. "*Entity1*" and "*Entity2*" correspond interacting proteins and are connected by "and". The head of "*Entity1* and *Entity2*" is "*Entity1*", and we restart a process from here. The PAS of "*Entity1*" is also an argument of the PAS of "between", so we step to "between". Because "between" is not a verb or a noun, we then step to another argument of "between" and it is the PAS of "interaction". "interaction" is a noun which is not interacting proteins, thus the process ends here. Finally, we make a PAS pattern that consists of shadowed PASs in the figure.

Pattern Division

For generalization based on the observation of class (2), we divide a pattern with only one verb into a verbal component and two nominal components. A verbal component consists of the verb and its next preposition if exists. Nominal components are the rest of the pattern, each for each interacting protein word. For the present implementation, we do not divide patterns with more than one verb and ones without any verbs.

Figure 7 illustrates an example. The verbal component consists of PASs of "interact" and "with". The PASs of "coreceptor" (1) and "region" (3) of the verbal component are connectors to nominal components and generalized to variables (*Xnoun1* and *Xnoun2*). And we mark "coreceptor" of the left nominal component as connectable to (unifiable with) *Xnoun* of a verbal component. Similarly, "region" of the right nominal component is marked as connectable.

Pattern Matching

To extract new protein-protein interactions, we match obtained patterns to parsing results of sentences in new input text. Because both patterns and parsing results of input sentences are represented in PASs, pattern matching is done by PAS matching: each PAS of the patterns is checked whether it is able to be matched (unified) to any PAS in the parsed sentences.

We use every combination of verbal and nominal components as a pattern. Only PASs of nominal components which are marked as connectable are unified (connected) to *Xnoun*s of verbal components. And we enumerate PASs of input sentences to resolve appositions in the same way with Pattern Extraction.

Result

We used Aimed [2] as a source tagged corpus for extraction rule construction and also a criterion of evaluation of IE by the constructed rules. Aimed we used consists of 199 Medline abstracts (1737 sentences[1]) tagged for both protein interactions and protein names. The abstracts were selected based on the Database of Interacting Proteins (DIP, [19]). Labels such as "TI - " and references are deleted from the sentences. We used the tags for protein names as already given in input, to separate the protein interaction extraction task from the protein name recognition task.

We measured accuracy of the IE task by two criteria: Word Unit and Abstract Unit. On Word Unit criterion, all word pairs corresponding to tagged interacting proteins have to be extracted (position-level measure). On Abstract Unit, a certain interacting pair has to be extracted from any of the sentences in the abstract (document-level measure).

[1] Among them, because of elapsed time, (i) we did not use 15 too long and/or complicated sentences for the rule construction (but considered them as failure cases in the evaluation) and (ii) did not try matching for other 10 sentences which are too complicated (treated them as failure cases in the evaluation).

Table 4: Accuracy of IE task

	Precision	Recall	F-measure
Word Unit	33.7%	33.1%	0.334
Abstract Unit	37.3%	45.3%	0.410

Resulting accuracy of the IE with the constructed rules is shown in Table 4. We made 10-fold cross validation, i.e. divided abstracts of Aimed into 10 sets, used each one set for evaluation and the rest for construction of extraction rules, and took average evaluation values of the 10 tests. The values are calculated as follows:

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}} \quad (1)$$

$$\text{Recall} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}} \quad (2)$$

$$\text{F-measure} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (3)$$

1. Positive: extracted as an interacting protein pair by the IE system.

2. Negative: not extracted

A. True: correct (i.e. true interacting proteins if extracted, not interacting proteins if not)

B. False: incorrect

Table 5 shows some of verbal and nominal components of frequent occurrence in one of training sets. Most of the verbal components, corresponding to classes (1) or (2), are usable as patterns by themselves (class (1)). "*Entity1* be *Entity2*" is an example of verbal components that become proper patterns only with proper nominal components.

Discussion

We investigated half (10 abstracts) of the result on one test set. The numbers of True/False Positive/Negative based on Word Unit and error causes are shown in Table 6. There were some major error causes, (A), (E), (F–H), (I), and also not frequent but substantial ones, (B), (C). Details of them and possible solutions are described in the following:

Most of the False Positives were caused by (A) incorrect patterns, such as "*Entity1* protein" and "*Entity1* complex", not as nominal components but acting as stand-alone patterns like "*Entity1* dimerization", a pattern of class (4). These patterns were constructed from incorrect parsing result or the present naive pattern extraction method which cannot extract adjective modifiers (classes (4c) and (5)). With little decrease of recall, we can easily eliminate the patterns by testing them on the corpus from which they are constructed: If a pattern extracts more False Positives than True Positives, it should be eliminated. Expected gain of precision is about 30% in maximum which would make our accuracy higher than the previous work with the same corpus [2].

Many False Negatives were (E) caused by parsing error, which can affect both of pattern construction and pattern matching. As the parser we used is now trained only on the general English corpus, the Penn Treebank, we can improve its accuracy by training it additionally on the GENIA Treebank.

In addition, there were also problems that (F–H) necessary pattern components did not occur in the corpus for the pattern construction. The reason why our recall result for the same precision is lower than the previous work [2] is that the corpus we used is slightly smaller than theirs and our PAS patterns are more precise on word relations than their surface word patterns allowing gaps. Further pattern generalization (such as generalization of "binding of *Entity1* and *Entity2*" and "binding of *Entity1* to *Entity2*") will gain more recall.

The lack of necessary patterns included (G1,H1) another kind of problem in which our method could not construct appropriate patterns with adjective modifiers (classes (4c) and (5)). An example of the case is for "*Entity1* have *Entity2* -binding property": From this example, our method constructed a verbal component "have" and a nominal component "property", but the appropriate nominal one is "-binding property". To solve this problem, distinction of common words ("have" and "property") and key words ("-binding") is needed.

For now, we consider (I) too complicated sentences (sentences with too many coordinations or parentheses) as IE failure cases because of elapsed time. These sentences were very few (10 in 1737 sentences), but might cause many False Negatives. Thus improvement in speed of IE (pattern matching) process is required.

On the other hand, there were a few but substantial False examples. One kind of them was (B) False Positives which required non-local information to distinguish correctly. An example is "binding of *EntityA* and *EntityB* to *EntityC*". There was a pattern "binding of *Entity1* and *Entity2*" which extracts correct interacting protein pairs, thus it extracted a pair (*EntityA*, *EntityB*). But for this example, correct pairs are (*EntityA*, *EntityC*), (*EntityB*, *EntityC*) based on a pattern "binding of *Entity1* to *Entity2*" and enumeration by a coordination. We need to introduce priority order on pattern matching, such as matching length order, to solve this problem. And there are other cases which require information in different sentences. To exclude these cases,

Table 5: Occurred Components of PAS Patterns

	Verbal Components		Nominal Components
37	*Entity1* interact with *Entity2*	15	domain$_{Xnoun}$ of *Entity1*
22	*Entity1* bind to *Entity2*	12	*Entity1* protein$_{Xnoun}$
19	*Entity1* be *Entity2*	8	*Entity1* complex$_{Xnoun}$
18	*Entity1* bind *Entity2*	7	ligand$_{Xnoun}$ for *Entity1*
10	*Entity1* interact *Entity2*	5	structure$_{Xnoun}$ of *Entity1*
10	*Entity1* associate with *Entity2*	5	region$_{Xnoun}$ of *Entity1*

Numbers denote occurrence times in a test corpus.

Table 6: Number of True/False Positive/Negative and Error Causes

True Positive	**21**
False Positive	**22**
(A) Incorrect Not-Divided Pattern	17
(B) Need Information of Other Parts/Sentences	3
(C) Need Negation Handling	1
(D) Test Corpus Error	1
False Negative	**44**
(E) Parsing Error	9
(F) Verbal Component Not Occurred in Training Corpus	5
(G) Nominal Component Not Occurred in Training Corpus	8
(G1) Need Further Extending	(2)
(H) Not-Divided Pattern Not Occurred in Training Corpus	13
(H1) Need Further Extending	(1)
(I) Too Complicated for Matching Process	8
(J) Representative Word Error for the Protein Name	1
(K) Test Corpus Error	3

One error may be caused by more than one cause, thus sums of error causes differ from the number of False Negative.

processing multiple sentences is required.

There was also a False Positive which required handling of a negative representation, "*Entity1* did not associate with *Entity2*". We need to gather words for such negative representations and refer to them in IE process.

Conclusion

We proposed to use predicate-argument structures (PASs) for automatic construction of patterns as IE rules. Because PASs abstract syntactical variants for the same information, patterns based on PASs are more generalized than those on surface forms of words. In addition, grounded on observation that most patterns can be divided into some components, we divided the patterns into components for generalization. On experiments of extraction of protein-protein interactions, we obtained 37.3% precision and 45.3% recall without any manual intervention. Extending pattern generalization to non-syntactical variations and filtering incorrect patterns by machine learning are planned.

Acknowledgements:
This work was partially supported by Grant-in-Aid for Scientific Research on Priority Areas (C) "Genome Information Science" from the Ministry of Education, Culture, Sports, Science and Technology of Japan.

Address for Correspondence:
Akane Yakushiji
University of Tokyo
Department of Computer Science
Hongo 7-3-1, Bunkyo-ku, Tokyo 113-0033, Japan

References

[1] Christian Blaschke and Alfonso Valencia. The Frame-Based Module of the SUISEKI Information Extraction System. *IEEE Intelligent Systems*, 17(2):14–20, 2002.

[2] Razvan Bunescu, Ruifang Ge, Rohit J. Kate, Edward M. Marcotte, Raymond J. Mooney, Arun Kumar Ramani, and Yuk Wah Wong. Comparative Experiments on Learning Information Extractors for Proteins and their Interactions. *Journal Artificial Intelligence in Medicine (Special Issue on Summarization and Information Extraction from Medical Documents)*, 2004.

[3] Nikolai Daraselia, Sergei Egorov, Andrey Yazhuk, Svetlana Novichkova, Anton Yuryev, and Ilya Mazo. Extracting Protein Function Information from MEDLINE Using a Full-Sentence Parser. In *Proc. the Sec-*

ond European Workshop on Data Mining and Text Mining for Bioinformatics, pages 15–21, 2004.

[4] Carol Friedman, Pauline Kra, Hong Yu, Michael Krauthammer, and Andrey Rzhetsky. GENIES: a natural-language processing system for the extraction of molecular pathways from journal articles. *Bioinformatics*, 17(Suppl. 1):S74–S82, 2001.

[5] GENIA Project. GENIA Treebank, 2004. http://www-tsujii.is.s.u-tokyo.ac.jp/GENIA/topics/Corpus/GTB.html.

[6] Minlie Huang, Xiaoyan Zhu, Yu Hao, Donald G. Payan, Kunbin Qu, and Ming Li. Discovering patterns to extract protein-protein interactions from full texts. *Bioinformatics*, 20(18):3604–3612, 2004.

[7] Jin-Dong Kim, Tomoko Ohta, Yuka Teteisi, and Jun'ichi Tsujii. GENIA corpus – a semantically annotated corpus for bio-textmining. *Bioinformatics*, 19(suppl. 1):i180–i182, 2003.

[8] Asako Koike, Yoshiyuki Kobayashi, and Toshihisa Takagi. Kinase Pathway Database: An Integrated Protein-Kinase and NLP-Based Protein-Interaction Resource. *Genome Research*, 13:1231–1243, 2003.

[9] G. Leroy and H. Chen. Filling Preposition-Based Templates to Capture Information from Medical Abstracts. In *PSB 7*, pages 350–361, 2002.

[10] Mitchell Marcus, Grace Kim, Mary Ann Marcinkiewicz, Robert MacIntyre, Ann Bies, Mark Ferguson, Karen Katz, and Britta Schasberger. The Penn Treebank: Annotating predicate argument structure. In *Proc. AAI '94*, 1994.

[11] MUC-6. Proc. the Sixth Message Understanding Conference (MUC-6), 1995.

[12] Toshihide Ono, Haretsugu Hishigaki, Akira Tanigami, and Toshihisa Takagi. Automated extraction of information on protein-protein interactions from the biological literature. *Bioinformatics*, 17(2):155–161, 2001.

[13] J. C. Park, H. S. Kim, and J. J. Kim. Bidirectional Incremental Parsing for Automatic Pathway Identification with Combinatory Categorial Grammar. In *PSB 6*, pages 396–407, 2001.

[14] J. Pustejovsky, J. Castan o, J. Zhang, M. Kotecki, and B. Cochran. Robust Relational Parsing Over Biomedical Literature: Extracting Inhibit Relations. In *PSB 7*, pages 362–373, 2002.

[15] Ivan A. Sag and Thomas Wasow. *Syntactic Theory*. CSLI publications, 1999.

[16] Kiyoshi Sudo, Satoshi Sekine, and Ralph Grishman. An improved extraction pattern representation model for automatic IE pattern acquisition. In *Proc. ACL 2003*, pages 224–231, 2003.

[17] Joshua M. Temkin and Mark R. Gilder. Extraction of protein interaction information from unstructured text using a context-free grammer. *Bioinformatics*, 19(16):2046–2053, 2003.

[18] Tsujii laboratory. Enju - A practical HPSG parser, 2005. http://www-tsujii.is.s.u-tokyo.ac.jp/enju/.

[19] I. Xenarios, E. Fernandez, L. Salwinski, X. J. Duan, M. J. Thompson, E. M. Marcottes, and D. Eisenberg. DIP: The database of interacting proteins: 2001 update. In *Nucleic Acids Research*, volume 29 (1), pages 239–241, 2001.

[20] Akane Yakushiji, Yusuke Miyao, Yuka Tateisi, and Jun'ichi Tsujii. Biomedical information extraction with predicate-argument structure patterns. In *Proc. the First International Symposium on Semantic Mining in Biomedicine*, 2005. to appear.

[21] Akane Yakushiji, Yuka Tateisi, Yusuke Miyao, and Jun'ichi Tsujii. Event Extraction from Biomedical Papers Using a Full Parser. In *PSB 6*, pages 408–419, 2001.

[22] Roman Yangarber, Ralph Grishman, Pasi Tapanainen, and Silja Huttunen. Automatic Acquisition of Domain Knowledge for Information Extraction. In *Proc. COLING 2000 - Volume 2*, pages 940 – 946, 2000.