Exploring Predicate-Argument Relations for Named Entity Recognition in the Molecular Biology Domain

Tuangthong Wattarujeekrit and Nigel Collier

National Institute of Informatics, 2-1-2 Hitotsubashi,
Chiyoda-ku, Tokyo, Japan 101-8430
tuangthong@grad.nii.ac.jp, collier@nii.ac.jp

Abstract. In this paper, the semantic relationships between a predicate and its arguments in terms of semantic roles are employed to improve lexical-based named entity recognition (NER) in the molecular biology domain. The semantic roles were realized in various sets of syntactic features used by a machine learning model to explore what should be the efficient way in allowing this knowledge to provide the highest positive effect on the NER. The empirical results show that the best feature set consists of *predicate's surface form, predicate's lemma*, *voice*, and the united feature of *subject-object head's lemma* and *transitive-intransitive sense*. The performance improvement from using these features indicates the advantage of the predicate-argument semantic knowledge on NER. There are still rooms to enhance NER by using this semantic knowledge (e.g. to employ other semantic roles besides *agent* and *theme* and to extend the rules for efficient identification of an argument's boundary).

1 Introduction

Named entity recognition (NER) is the task aiming to identify and categorize entities appearing in text. According to the Message Understanding Conferences (MUCs) [1], it is the lowest level in the task hierarchy of Information Extraction (IE) system. The entities to be recognized in the newswire domain include persons, organizations, locations, email addresses, and so on, whereas in the molecular biology domain, molecular entities such as genes, proteins, small molecules, chemical molecules, tissues, etc. need to be recognized. Not only is NER an important component of molecular biology IE to reach the goal of discovering biological pathways, but it is also beneficial to other applications of biological text mining. For instance, document retrieval where a relevant subset of documents are obtained [2] and document clustering where similar documents are grouped together [3]. For example, after NER has been used to process the sentence "*Cytokines bind to hematopoietin receptors and activate JAK kinases*", the fact that *Cytokines, hematopoietin receptors* and *JAK kinases* are referred to three different types of protein would be extracted. The different focus among researches gives variety to the granularity of concept classes to be distinguished. For example, to work with the GENIA ontology, 36 biologically nominal categories needed to be grouped [4].

Although, NER in the molecular biology domain has received wide scale attention by many researchers for nearly a decade, the overall performance is still far from human's capability [5-12]. As can be seen from the most recently shared-task of NER in the molecular biology domain (JNLPBA-2004), the best performance is only 72.6 for F-measure [9]. Contrastingly, the accuracy in general news-based NER is about 96% in MUC-6 [1] which is at near human levels of performance. This lag should mainly be due to the lack of naming convention[1] which leads to several sources of difficulties for NER. This work aims to handle two main difficulties as follows. First, the difficulty results from terminological variations i.e. molecular names may be formed by using a standard English word (e.g. "*light*", "*map*", "*complement*") or using an amino acid sequence (e.g. "*amino acids [aa] 1 to* 25") or using alpha numeric (e.g. "*9-cis retinoic acid*"). Second, the difficulty is from polysemy which is the ambiguity of a name that can refer to two or more different entities. Polysemy is classified into two cases: homonymy and systematic polysemy. Homonymy relates to the ambiguity of a name referring to unrelated meanings or objects (e.g. the term "*cat*" can refer to "*choline acetyltransferase protein*" and "*catalase gene*"). Systematic polysemy relates to the ambiguity of a name referring to the objects which systematically relate to each other (e.g. the term "*BCL-6*" can refer to "*B-cell CLL/lymphoma 6 gene*" and its protein product). These difficulties are expected to increase when we scale-up NER from an abstract to full text. Thus, most molecular NER systems now take place on MEDLINE abstracts.

In this paper, we argue that to overcome the limits in what can be achieved by existing NER systems traditionally based on lexical features and context features derived from neighboring words [7, 10-12], deeper knowledge such a predicate-argument relationship should be taken into account. This hypothesis is motivated by the basic observation that events are realized as predicates[2] and their participating named entities (NEs) as the predicates' arguments. The semantic role each argument plays in the event should impose type restrictions on the entity within the argument. The investigation of how to efficiently transform the knowledge of predicate-argument relations into features of training data for our NER system using a machine learning approach is the main focus in this work.

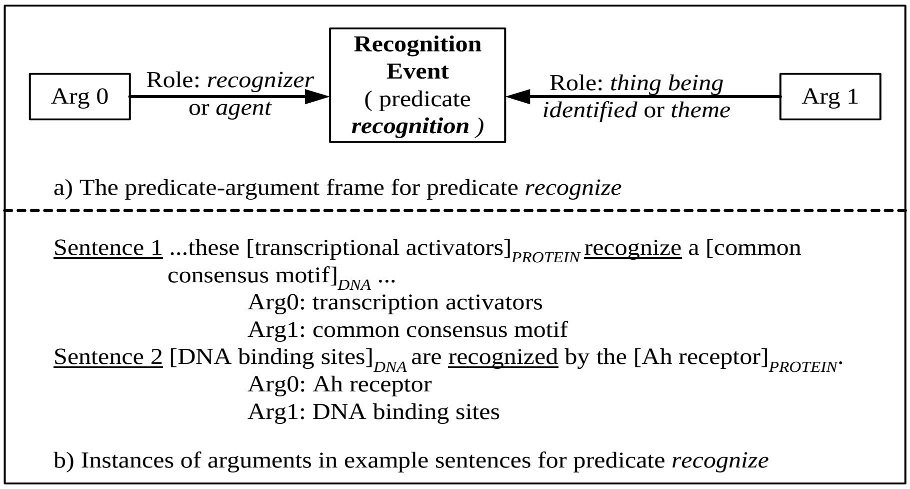
The paper is organized as follows. Section 2 discusses how predicate-argument relation is useful to NER and how other researchers have taken efforts to apply this knowledge. Section 3 outlines the transformation of predicate-argument relations into our machine learning features. Section 4 shows experimental results and the analysis on the results. Section 5 discusses concerning impediments to high performance improvement. Finally, Section 6 summarizes the conclusion.

[1] Some efforts have been shown to standardize in naming biological entity (e.g. Guidelines of Human Gene Nomenclature, Drosophila Gene Nomenclature, etc., however many biologists often do not follow the recommended nomenclature.

[2] Hence, a predicate refers to a verb which can exist in a sentence in its verbal form (e.g. infinitive – *to activate*, present simple – *activate* or *activates*, past simple – *activated*, present or past participial – *activating* or *activated*), or its nominal form (e.g. *activation*).

2 Predicate-argument Relation and Biological NER

A frame of predicate-argument structure (PAS) represents a set of semantic relationships in terms of the specified role each argument plays in the event indicated by a predicate. For example, the predicate-argument frame of the predicate *recognize* which is used to express the recognition event in the molecular biology domain would be as Fig. 1(a). Thus, deeper knowledge than surface syntax of sentence 1 and 2 can be obtained as shown in Fig. 1(b). That is the occurrence of a recognition event would be participated by two participants (i.e. Arg0 and Arg1). The first argument (Arg0) has a relationship to the predicate *recognize* as a *recognizer* or *agent* of the event and the second argument (Arg1) plays role as *thing being identified* or *theme* in the event. Sentence 1 shows the usage of predicate *recognize* in active voice. The sentence's surface subject which is "*transcriptional activators*" plays role as *agent* and its surface object "*common consensus motif*" plays role as *theme*. On the contrary, a surface subject of sentence 2 which is "*DNA binding sites*" plays role as *theme* and a surface object "*Ah receptor*" plays role as *agent* as the predicate *recognize* is used in *passive voice*.



a) The predicate-argument frame for predicate *recognize*

Sentence 1 ...these [transcriptional activators]$_{PROTEIN}$ recognize a [common consensus motif]$_{DNA}$ ...
Arg0: transcription activators
Arg1: common consensus motif

Sentence 2 [DNA binding sites]$_{DNA}$ are recognized by the [Ah receptor]$_{PROTEIN}$.
Arg0: Ah receptor
Arg1: DNA binding sites

b) Instances of arguments in example sentences for predicate *recognize*

Fig. 1. The semantic relationships between predicate *recognize* and its argument

As can be noticed from Fig. 1, the argument playing role as *agent* belongs to class *PROTEIN* in both sentences. Similarly, the argument with semantic roles of *theme* belongs to class *DNA*. This restriction of NE-types corresponding to arguments' semantic roles is a key concept to employ semantic relations in PAS for enhancing molecular NER system.[3] As the NER system used in this work is based on Support

[3] The empirical evidence observed on GENIA V3.02 corpus (http://www-tsujii.is.s.u-tokyo.ac.jp/~genia/topics/Corpus/) shows that the frequency of occurrence for *PROTEIN* to be *agent* in an recognition event is about 53% and for *DNA* to be *theme* is about 26%.

Vector Machines (SVMs) [13], this predicate-argument relationship knowledge is required in the form of machine learning features.

Recently, due to the ability of PAS to straightforwardly represent the biological event, this knowledge has been used mostly as a reference frame to extract instances of biological events from text, e.g. the protein-protein interaction event [14-17]. To our knowledge, two previous works have shown the efforts to employ this knowledge for NER in the molecular biology domain [6, 8]. In the first approach [6], the verb complementation patterns between each verb and the arguments which their concept classes are known have been automatically learnt by using an iterative reasoning process based on a partial order relation induced by the domain-specific ontology. Then, an unknown class term will be classified to the potential class based on the similarity measure between this new term's verb complementation patterns and the pre-analyzed known class term. This method still gets low performance to classify terms related to the small set of verbs that were studied (i.e. F-measure = 40.68%, 26.28%, 21.85%, and 19.69% for *bind*, *inhibit*, *interact*, and *mediate* respectively). In the second approach [8], a set of verbs, such as *inhibit*, *express*, *bind*, and *activate* has been set as binary features in HMM-based model. Unexpectedly, the overall F-measure has decreased by 1.8. One possible explanation for this result is that it could be due to the impractical way to represent predicate-argument relations in the model. The verb features represented only the knowledge that the verb exists in the context of the term or not.

In this paper, we explore an efficient way to exploit the semantic relations between predicate and its argument for improving SVM-based NER system.

3 Our Method

Our SVM-based NER system develops from the learning model of Takeuchi and Collier [7] in which the Tiny SVM[4] with the multi-class strategy of one-against-one was used. The context window was set to ± 1 providing features for the previous word, current word, and next word. Also, the two previous class assignments were taken into the model. The training data used in our system is in a form of a column formatted table of features with the NE classes provided in IOB2 format[5]. We form 6 sets of features (i.e. the Model 1 – Model 6) to be trained by SVMs. Model 1 contains only lexical-based features proposed in earlier studies to reduce known problems of ambiguity for term recognition. This model is used as a base model to be compared with the Model 2-6 in which predicate-argument related features are included in addition to lexical-based features. Thus, the significance of the semantic relationships represented in PAS to NER system can be evaluated. In order to evaluate the efficiency of different ways to convert this semantic knowledge into features of input data, Model 3, 4, 5, and 6 will be compared to the Model 2. How each feature set is derived and what thought is underlying the forming of it will be explained in section 3.3.

[4] The Tiny SVM package is available from http://chasen.org/~taku/software/TinySVM/.

[5] IOB2 format is a standard format for word-chunk. The tag "O" is given to words outside a chunk, "B-*k*" to the first word in a chunk of type *k*, and "I-*k*" to the remaining words.

3.1 Data Set

The GENIA corpus V3.02, the largest annotated corpus in the molecular biology domain available to public, is used as our data set of the NE tagged text. As the predicate-argument relationship is a specific characteristic for each individual predicate, we decide to explore the influences of features derived from the knowledge of predicate-argument relation separately for each predicate. In this paper, we mainly focus to a predicate in verbal form, thus a collection for each predicate will be retrieved from GENIA by using the criteria that the relevant sentences must contain a focus predicate in verbal form at least once. With regard to the classes of NE used in evaluation, we follow the JNLPBA-2004 shared task [9] to use the conflated set of classes consisting of 5 classes: protein, DNA, RNA, cell line, and cell type.

3.2 Selection of Predicates to Be Explored

We started selecting predicates by gathering predicates used in earlier works to capture biological events [14-16] and predicates used in our previous work to construct the PASBio[6] resource [18]. Most predicates from the 44 predicates which have been gathered are found rarely in the GENIA corpus. In order to avoid having too small set of training data, we filtered out predicates containing less than 100 examples[7]. This filtering process results in a set of 19 predicates in which *bind* has a biggest and *alter* has a smallest volumes of training data, i.e. 825 and 102 examples respectively.

Due to our intuition that the proportion of belonging to a NE class of an *agent* argument and a *theme* argument[8] should be a key impact to the performance of NER system when predicate-argument related features are applied, we selected 6 predicates from the total 19 predicates to be the representative predicates of the 3 groups as follows. First, the predicates *encode* and *recognize* were selected to be the representatives for a group of predicates having arguments both *agent* and *theme* with higher possibility to belong to a NE class than non-NE class. Second, the predicates *block* and *lead* were selected for a group of predicates having arguments both *agent* and *theme* with lower possibility to belong to a NE class than non-NE class. Third, the predicates *regulate* and *associate* were selected for a group of predicate having arguments either *agent* or *theme* with higher possibility to belong to a NE class than non-NE. Table 1 shows the proportion of the arguments of these representative predicates to 5 classes of NEs.

[6] PASBio resource contains frames of predicate-argument structure analyzed from the literatures in MB domain. Available online at http://research.nii.ac.jp/~collier/projects/PASBio/.

[7] The number of examples is a number of clauses containing a particular predicate. In a sentence, it is possible to have more than one clause related to the predicate in focus.

[8] The *agent* argument refers to the argument which has syntactic role as *subject* in the case of active voice and refers to the argument having syntactic role as *object* introduced by the preposition “by” in the case of passive voice. The *theme* argument refers to the argument which has syntactic role as *object* in the case of active voice and refers to the argument having syntactic role as *subject* in the case of passive voice.

Table 1. The proportion of *agent* and *theme* arguments to 5 classes of NEs.

Predicate		Group 1		Group 2		Group 3	
		Agent and theme with High NE%		Agent and Theme with Low NE%		Only Agent or Theme with High NE%	
		encode	recognize	block	lead	regulate	associate
Agent Argument	Total Agent	228	113	209	241	381	39
	Protein%	03.51	53.10	28.71	06.64	54.33	41.03
	DNA%	47.81	00.00	02.39	00.41	08.92	00.00
	RNA%	04.82	00.00	00.00	00.00	00.00	00.00
	Cell line%	00.44	07.08	00.48	01.24	00.00	05.13
	Cell type%	00.44	14.16	00.48	00.83	01.05	05.13
	Total NE%	57.02	74.34	32.06	09.13	64.30	51.29
	Non-NE%	42.98	25.66	67.94	90.87	35.70	48.71
Theme Argument	Total Theme	234	94	234	296	482	614
	Protein%	66.67	25.53	11.54	02.36	10.17	16.61
	DNA%	00.85	25.53	01.71	00.68	10.17	05.05
	RNA%	00.85	00.00	00.85	00.00	00.21	00.00
	Cell line%	00.00	00.00	00.00	00.00	00.41	00.49
	Cell type%	00.00	00.00	01.28	00.34	00.41	01.95
	Total NE%	68.37	51.06	15.38	03.38	21.37	24.10
	Non-NE%	31.63	48.94	84.62	96.62	78.63	75.90

Moreover, if the number of examples for predicates from each group is not in balance, it could be difficult to compare their results. The intention to balance the number of examples of predicates to be investigated had been applied for selecting these representative predicates as well. More precisely, these 6 predicates were selected because they also conform to the condition that they have the numbers of examples nearly the average value for the total 19 predicates.

3.3 Derivation of Feature Sets

The Conexor FDG parser [19] which is widely used and is considered to be a state-of-the-art commercial parser is used to parse our NE tagged text. In addition to each word's morphological information (i.e. surface form and lemma form) and lexical category (i.e. part-of-speech), this parser also provides functional dependency relations between words which is one of a key syntactic information for acquiring semantic relationships between a predicate and its arguments. These parsing results are used to derive a set of features used in the Model 1-6 as follows.

Model 1. This model composes of 6 features widely recognized as important for NER task. These features include *surface word*, *lemma form*, *head word of noun phrase*, *part-of-speech*, *orthographic feature*, and *phrase-chunk*. This model is named lexical-based model as it is based mainly on lexical information. As stated before, this model is used as a base model for evaluating the importance of the semantic knowledge represented in PAS to NER system.

Model 2. This model contains all lexical-based features used in the Model 1, with additional set of features constituted from syntactic information to represent arguments' semantic roles). These supplementary features consist of *predicate* *surface form*, *predicate lemma*, *voice* and *surface syntactic role*. The *voice* feature is used to distinguish between *active* and *passive* voice of the predicate. The tag "*ACT*" represents *active* voice and "*PAS*" represents *passive* voice. The *surface syntactic role* feature describes syntactic functions (i.e. *surface subject* or *surface object*) of the head word of a noun-phrase which is bound as the predicate's argument. Tags used are "*SSUBJ*" for *surface subject* and "*SOBJ*" for *surface object* which is found as direct object. Moreover, the tag "*PCOMP*" used for *surface object* which is found as a prepositional complement. For instance, from sentences "A binds B." and "A binds to B.", "A" will be tagged with "*SSUBJ*" in both sentences but "B" will be tagged with "*SOBJ*" for the former sentence and "*PCOMP*" for the latter. The procedures used to identify the argument's boundary are illustrated in section 3.4. The semantic roles of arguments can be determined partially from a combination of the 4 additional features used in this model. Only if both *surface subject* and *surface object* co-occur with a target verb, the argument with syntactic function as *subject* and the argument with syntactic function as *object* will be confidently concluded that they semantically plays role as *agent* and *theme* respectively in case of *active* voice and vice versa in case of *passive* voice. The correct determination of semantic role would lead to the correct NE classification; underlying our hypothesis that semantic relationships in PAS (arguments' semantic roles) for each predicate confine classes of NEs participating the event indicated by the predicate. However, as the arguments with the same semantic role possibly belong to different NE classes, the lexical-based features and semantic relationships are required altogether to solve this ambiguity. This model is a PAS-based model which will be extended to the Model 3-6 by adding features of several kinds of syntactic information in order to decrease the ambiguity in determining semantic roles.

Model 3. Path feature representing the syntactic path from the subject argument to the related predicate and from the related predicate to the object argument is added to all features used in Model 2. The path is derived from the flat structure of dependency tree resulting from the parser. For example, the path between the subject constituent and the predicate is "NP_VP_ADVP_VP" and the path between the object constituent and the predicate is "VP_PP_NP" for the sentence "[Increased cytokine secretion]$_{\text{NP}}$ [was]$_{\text{VP}}$ [specifically]$_{\text{ADVP}}$ [inhibited]$_{\text{VP}}$ [by]$_{\text{PP}}$ [G1]$_{\text{NP}}$".

Model 4. A feature representing a pair of subject and object's heads is added to the Model 2 instead of path feature. This feature is designed following the intuition that a NE class of an agent should restrict a possible type of a NE playing role as theme and vice versa. The using of a subject-object head pair in lemma form would help to reduce data sparseness problem compared to the using in surface form. For the sentence in Fig. 2, the subject-object head feature will be "compound_complex".

Model 5. This model augments the Model 2 with a feature representing if a predicate is used in transitive or intransitive sense. For each surface subject's constituent, a tag "fobj" is set if the surface object is found in the current clause. A tag "O" is set if the surface object is not found. However, this feature helps just in part to correctly determine transitive or intransitive sense implicit in the usage of a predicate. It is due to the object argument can be omit in a clause although a predicate is used in transitive sense. For instance, the predicate “eat” is used in transitive sense without mentioning an object in the sentence “Yesterday, John ate at ABC restaurant”.

Model 6. This model is considered as a joining of the Model 4 and the Model 5. A pair of subject and object’s heads is used to be assigned to a column of transitive-intransitive feature instead of “fobj” when the object is found in the clause.

The lexical-based features used in Model 1 will be given to every word or token in a sentence. Contrastingly, the PAS-related features proposed in Model 2-6 will be assigned to only the constituents bound as the arguments having syntactic function as *surface subject* and *surface object* of the focus predicate. How to identify the boundary of these constituents is as follows.

3.4 Sub-structure Recognition

The sub-structure recognition is the process to identify the tokens that constitute arguments of predicates. In our study, we have focused mainly on a predicate in verbal form but not nominal form. Therefore, for a predicate such as *activate*, the surface forms of this predicate to be analyzed include *activate*, *activates*, *activated*, and *activating*¸ but not *activation*. Furthermore, only an argument corresponding to the syntactic relation of either subject or object is bound in this study. At present, there is a lack of practical semantic role labeling systems to identify arguments of a predicate, especially for the molecular biology domain. Thus, this study which is to investigate the constitution of semantic relationship between predicate and its arguments simplifies its scope to arguments as grammatical subject or object.

The algorithm used to find a subject constituent and an object constituent of each predicate is based mainly on the functional dependency relations between words obtained from the parser as shown in Fig. 2. It comprises of several steps as follows. First, find a position of target predicate which must be in a verbal form. Second, interpret the verb’s voice by checking at the column *Surface Syntactic* (Fig. 2, C. 6) of the verb token (Word No. 3). If it is *%VA*, the verb is an active verb. On the other hand, if it is *%VP*, the verb is a passive verb. Third, find a token functioning as a subject or object of the target verb by traversing through syntactic relations given by the parser (Fig. 2, C. 4). Basically, the system will traverse up until *subj:>#* is found in case of subject and traverse down until *obj:>#* is found in case of object.[9] From Fig. 2, the token *compounds* is found to have subject relation to the verb *alter* and the token *complex* is found to be an object. Subsequent to founding the head of subject or object, the full boundary of a subject or an object is identified by propagating to the premodifiers of a noun which is a subject head or an object head. These premodifiers will have *@A>* at the column *Functional Tag* in parsing data (Fig. 2, C. 5). All modifiers except determiners are included in surface subject or surface object boundary as shown in Fig. 2 that *NFAT-1* and *transcriptional* are included but *the* is not included in the boundary of surface object containing *complex* as the object head. A determiner is not included into both boundary of object and subject because any determiners

[9] Hence, the symbol # refers to the word number of the target verb.

never ever are parts of the biological terms. This rule not to include a determiner is also used by Rindflesch and colleagues to extract binding relationships [17].

C. 1	C. 2	C. 3	C. 4	C. 5	C. 6	C. 7	
Word No.	Surface Form	Lemma Form	Syntactic Relation	Functional Tag	Surface Syntac-tic	Part-of-Speech	
1	Both	Both	det:>2	@DN>	%>N	DET -	
2	compounds	compound	subj:>3	@SUBJ	%NH	N NOM_PL	Subject
3	altered	Altered	main:>0	@+FMAINV	%VA	V PAST	Verb
4	the	The	det:>7	@DN>	%>N	DET -	
5	NFAT-1	NFAT-1	attr:>6	@A>	%>N	N NOM_SG	Object
6	transcrip-tional	transcrip-tional	attr:>7	@A>	%>N	A ABS	
7	complex	Complex	obj:>3	@OBJ	%NH	A ABS	
8	,	,					
9	causing	Causing	ha:>3	@-FMAINV	%VA	V ING	
10	its	Its	attr:>11	@A>	%>N	PRON GEN_SG 3	
11	retarded	retarded	attr:>12	@A>	%>N	A ABS	
12	mobility	Mobility	obj:>9	@OBJ	%NH	N NOM_SG	
13	on	On	loc:>9	@ADVL	%EH	PREP -	
14	gels	Gel	pcomp:>13	@<P	%NH	N NOM_PL	
15	.	.					

Fig. 2. Boundaries of surface subject and object of the verb *alter* recognized by the system (*thick squares*) using the FDG parsing result of a sentence “Both compounds altered the NFAT-1 transcriptional complex, causing its retarded mobility on gels.”

To look for *subj:>#* or *obj:>#*, at the column *Syntactic Relation* (Fig. 2, C. 4), to get a subject head or an object head is practical for a simple clause. In some cases, a token holding *subj:>#* or *obj:>#* is not found as a subject head or an object head has a direct dependency relation to another token but not to a target verb. The more complex criterion needs to be processed to recover the relations between a subject and an object to the target verb. These cases are as follows: 1) an auxiliary verb (e.g. be, do, have, etc.) or a verb phrase functioning similar to auxiliary verb (e.g. play a role in, is required to, have been shown to, etc.) precedes a target verb, 2) a target verb shares its subject or object with other verbs, 3) a target verb is a main verb in a subordinate clause of which the relative pronoun presents as the subject, and 4) an object of a target verb is introduced by a preposition following a target verb[10].

[10] Due to the space limitation, the details of the extended criterion for these complicated cases to identify the boundaries of subject and object arguments cannot be explained here.

4 Experimental Results and Analysis

All results reported here are given as F1-scores calculated using 10-fold cross validation. F1-score is defined as *F1= (2PR)/(P+R)* where *P*, called as *Precision*, is the ratio of the number of correctly found NE chunks to the number of found NE chunks and *R*, called as *Recall*, is the ratio of the number of correctly found NE chunks to the number of true NE chunks.

Table 2. F1-scores of the 6 representative predicates trained with features in Models 1-6

Predicates	Group 1		Group 2		Group 3	
	Agent and Theme with High NE%		Agent and Theme with Low NE%		Only Agent or Theme with High NE%	
Model	encode (265)	recognize (121)	block (270)	lead (288)	regulate (525)	associate (377)
Model 1 (*Lexical-based*)	56.60	47.24	51.19	57.01	61.87	52.09
Model 2 (*PAS-based*)	**57.56**	**49.39**	**51.47**	**57.40**	60.48	51.48
Model 3 (*Path*)	**58.38**	**48.47**	**52.23**	56.70	60.13	51.29
Model 4 (*Head Pair*)	**57.16**	**49.54**	**51.85**	**57.12**	60.72	50.43
Model 5 (*Trans/Intrans*)	**57.69**	**49.16**	**52.02**	**57.53**	60.01	51.40
Model 6 (*M4+M5*)	**57.64**	**49.39**	**51.95**	**57.49**	60.37	50.97

The results of 6 predicates using the feature sets from the Models 1-6 are shown in Table 2. In each column, the F1-score of a corresponding predicate is given for Model 1 (Lexical-based model), Model 2 (PAS-related model), Model 3 (the Model 2 added with Path feature), Model 4 (the Model 2 added with Pair of subject and object's heads feature), Model 5 (the Model 2 added with Transitive/Intransitive feature) and the Model 6 (the Model 4 is embodied into the Model 5). For each predicate, the higher F1-scores from the models which outperform the Model 1 are shown in *bold* number. The models with *bold* number indicate the positive effect of PAS-related features to NER. Moreover, if the F1-scores in any models among Models 3-6 are higher than in Model 2, the scores will be highlighted with *gray* background. This helps to notice which PAS-related feature in addition to features used in PAS-based model (Model 2) has capability to increase positive effect of semantic relations between predicate and its arguments.

As can be observed from Table 2, the simple representation of PAS-related knowledge such in Model 2 improve the performance for all predicates except the predicates *regulate* and *associate* which have only *agent* or *theme* argument with higher possibility to belong to a NE class than non-NE. Moreover, these Group 3's predicates do not show any improvement in other models using PAS-related features (Model 3-6) compared to the lexical-based model (Model 1). Therefore, they will not be covered in the following discussion of how the extra PAS-related features used in Models 3-6 help to improve the performance of PAS-based features used in Model 2.

With regard to Path feature (Model 3), the performance is improved for only the model training on data set of predicate *encode* and *block*. Empirically, one reason we found for this is the surface subject and surface object of these two predicates are located close to the predicate in most of the cases. For example, the path patterns between arguments and the predicate *encode* of "…[proteins]$_{\mathrm{NP}}$ [encoded]$_{\mathrm{VP}}$ [by]$_{\mathrm{ADVP}}$

[these two latter genes]$_{NP}$…" are "NP_VP" for the subject argument and "VP_ADVP_NP" for the object argument. Due to short path patterns, so the path patterns can be generalized throughout the data sets. On the contrary, long path patterns are mostly found in the samples of other predicates (i.e. *recognize* and *lead*). For example, from the sentence "[Control peptides]$_{NP}$ [corresponding]$_{VP}$ [to]$_{ADVP}$ [the normal pml]$_{NP}$ [and]$_{O}$ [RAR alpha proteins]$_{NP}$ [were]$_{VP}$ [not]$_{ADVP}$ [recognized]$_{VP}$.", the path from the subject argument "Control peptides" to the predicate *recognize* is "NP_VP_ADVP_NP_O_NP_VP_ADVP_VP". This long path pattern would causes data sparseness problems for the path feature.

The next feature, the Head Pair feature, does not show its usefulness for predicates *encode* and *lead*. The reason for the predicate *lead* is that its arguments both as *agent* and *theme* are prone to be non-NE rather than to belong to NE class, thus the pair of its arguments' head words can have many variants. It causes this feature ineffective to constrain NE functioning as subject with NE functioning as object and vice versa. In case of predicate *encode*, although both arguments of it are prone to belong to NE classes rather than to be non-NE, the Head Pair feature does not show its positive effect. As the predicate *encode* used in the molecular biology domain has its specific meaning to describe relationships between genes and gene products, the head pair of arguments for this predicate is mostly found as *gene_protein*. Therefore, this feature contains too general information to be helpful for *encode*.

In case of Transitive/Intransitive feature, we believe that this feature should be useful to improve performances of all predicates. This feature is important to correctly interpret semantic role of an argument. For instance, the subject in the sentence "John broke the window" has the semantic role as *agent* but the subject in the sentence "The window broke" has semantic role as *theme*. These two sentences illustrate that to know only syntactic function as subject or object cannot have a correct determination on semantic role. The difference between these two sentences is that the predicate *break* is used in transitive sense in the former sentence and intransitive sense in the latter. Therefore, to give information stating if the object is found in a sentence or not would help to some extent to imply sense in which the predicate is used. The performance of the model having this feature (Model 5) should outperform the PAS-based model (Model 2). However, the performance for *recognize* has decreased when this feature is applied. From our analysis, the problem originates from parsing error of failing to provide syntactic relations between words. For instance, the FDG parser fails to give the constituent "DNA binding sites" syntactic relation as the object of "recognizes" in the sentence "The Ah receptor recognizes DNA binding sites for the B cell transcription factor" This causes subsequent problem to the Transitive/Intransitive feature, i.e. this feature is set to "O" to represent that the predicate *recognize* is used in intransitive sense, whereas it does not. This incomplete parsing result accounts for decreasing F1-score of *recognize* when using the Transitive/Intransitive feature (Model 5) compared to when not using it (Model 2).

In order to evaluate the contribution of PAS-related features from different models, the average F1-score from each PAS-related model (Model 2-6) is compared to the average F1-score of the lexical-based model (Model 1). Without considering the mix model (Model 6), the results show that the Transitive/Intransitive feature (Model 5) gives the highest contribution as expected. Some more improvement can be obtained in Model 6 when the Head Pair feature (Model 4) is embedded in the Transitive/Intransitive feature (Model 5). Thus far, the Model 6 is considered to be the best model in this work with the improvement, on average, in F1-score of 1.11 as shown in Table 3. Furthermore, each predicate reflects the benefit from using PAS-related features in different levels of improvement, listed in descending order as *recognize*, *encode*, *block*, and *lead*.

Table 3. The improvement in F1-scores of Model 6 (the best of PAS-related model) compared to Model 1 (the lexical-based model)

Predicates	Number of Examples	Model 1 (*Lexical-based*)	Model 6 (*M4+ M5*)	Improvement
Encode	265	56.60	57.64	1.04
Recognize	121	47.24	49.39	2.15
block	270	51.19	51.95	0.76
lead	288	57.01	57.49	0.48
Average of Improvement				1.11

5 Discussions

In Table 3, the experimental results have shown that the PAS-related features make only small progress in NER. However, it is not because semantic relationship between predicate and its argument is not an important knowledge to improve lexical-based NER. The incorrect identification of an argument boundary is an impediment for the system to acquire the actual performance improvement. This impediment is mainly caused by a failing of parser to provide syntactic relation information between tokens. One of its examples has already been shown in the previous section to explain why the Transitive/Intransitive feature degrades the performance of *recognize*. To investigate the contribution of PAS-related features without the impact from parsing error, the arguments *agent* and *theme* are identified manually on training examples of predicates *recognize* and *encode* (100 examples for each predicate). These two predicates are selected for this experiment because they obtain higher performance improvement than other predicates. The 2 sets of training data are trained by using features in Model 1 and Model 6 to calculate the performance improvement. The predicate *encode* obtains performance improvement of 2.40 from training on only 100 manual-examples (about 38% of parsing-examples)[11]. This performance improvement is about 2 times of what obtained from 265 parsing-examples (Table 3). In case of predicate *recognize*, from training on 100 manual-examples, the performance improvement increase to 6.12 which is about 3 times of what is obtained from 121 parsing-examples. The size of manual-examples of *recognize* is nearly equal to the parsing-examples' size, thus it can be implied that the parsing error can decrease the performance improvement at least 3 times.

[11] Hence, the training examples are called manual-examples when argument boundaries are identified manually and are called parsing-examples when argument boundaries are identified automatically based on syntactic relation information given by the parser.

In addition to the parsing error, the more complex rule to identify an argument boundary is required for some specific cases. For example, the constituent “multiple isotypes” in the sentence “T cells express multiple isotypes of protein kinase C” will be bounded to be *theme* argument of predicate *express* after the general algorithm for sub-structure recognition is applied. However, the real argument playing semantic role as *theme* which is related to NE-type protein is the constituent “protein kinase C”. Therefore, a set of rules to distinguish between a quantifier (e.g. “multiple isotypes”) and a real argument (e.g. “protein kinase C”) is required. Moreover, a rule set to include or not to include an entity’s abbreviation name (always mentioned in a bracket) in an argument boundary is required as well. For instance, in GENIA corpus V3.02 the constituent “cytokine receptor gamma chain (gamma c) gene” of a sentence “…cytokine receptor gamma (gamma c) gene encodes a component of …” is hand-annotated as one named entity, but the constituent “Sterol regulatory element (SRE)” of a sentence “….Sterol regulatory element (SRE) has been recognized …” is separated into two named entities (i.e. “Sterol regulatory element” and “SRE”).

In order to allow semantic knowledge of predicate-argument relationship covering semantic roles of *agent* and *theme* to express its actual contribution, the sources of errors in identifying an argument boundary as mentioned above must be handled.

6 Conclusions

In this work, we have shown that the deeper knowledge of semantic relationship between a predicate and its argument is benefit for NER. The choice of syntactic features to represent the PAS semantic knowledge is the key issue underlying the efficient employment of this knowledge. So far, the best set of syntactic features consists of features *predicate’s surface form*, *predicate’s lemma*, *voice*, and the united feature of *subject-object head’s lemma* and *transitive-intransitive sense*. The highest improvement is found from applying these features to the training examples of predicate *recognize*. Without parsing error which is one of the problems that can impede the contribution of the predicate-argument semantic knowledge to NER system, the highest improvement for *recognize* can reach to 6.12 F1-score.

Besides dealing with an argument’s boundary identification discussed in this work, there are still rooms to enhance NER by using this PAS knowledge such as employing syntactic features to represent other semantic roles in addition to *agent* and *theme*.

References

1. DARPA. The 6th Message Understanding Conference. Columbia, Maryland (1995)
2. Stapley, B. J., Benoit, G.: Biobibliometrics: Information retrieval and visualization from co-occurrences of gene names in Medline abstracts. Pac. Symp. Biocomp.(2000) 529-540
3. Willett, R.: Recent trends in hierarchic document clustering: a critical review. Information Processing & Management. (1998) 25: 577
4. Ohta, T., Tateishi, Y., Kim, J. D.: The GENIA corpus: An annotated research abstract corpus in the molecular biology domain. HLT. (2002)

5. Fukuda, K., Tamura, A., Tsunoda, T., Takagi, T.: Toward information extraction: identifying protein names from biological papers. Pac. Symp. Biocomp. (1998) 707-718
6. Spasic, I., Nenadic, G., Ananiadou, S.: Using domain-Specific Verbs for Term Classification. The ACL Workshop on NLP in Biomed. (2003) 17-24
7. Takeuchi, K., Collier, N.: Use of Support Vector Machines in Extended Named Entity Recognition. CONLL. (2002) 119-125
8. Zhou, G., Su, J.: Exploring Deep Knowledge Resources in Biomedical Name Recognition. The Joint Workshop on NLP in Biomed. and its App. (JNLPBA). (2004) 84-87
9. Kim, J. D., Ohta, T., Tsuruoka, Y., Tateishi, Y., Collier, N.: Introduction to the Bio-Entity Task at JNLPBA. (2004) 70-75
10. Collier, N., Nobata, C., Tsujii, J.: Extracting the names of genes and gene products with a Hidden Markov Model. COLING. (2000) 201-207
11. Kazama, J., Makino, T., Ohta, Y., Tsujii, J.: Tuning Support Vector Machines for Biomedical Named Entity Recognition. The ACL Workshop on NLP in Biomed. (2002) 1-8
12. Lee, K. J., Hwang, Y. S., Rim, H. C.: Two-phase biomedical NE Recognition based on SVMs. The ACL Workshop on NLP in Biomed. (2003) 33-40
13. Vapnix, V. N.: The Nature of Statistical Learning Theory. Springer-Verlag, New York (1998)
14. Blaschke, C., Andrade, M. A., Ouzounis, C., Valencia, A.: Automatic extraction of biological information from scientific text: Protein-protein interactions. The Int. Conf. on Intelligent System Molecular Biology. (1999) 60-67
15. Ono, T., Hishigaki, H., Tanigami, A., Takagi, T. Automated extraction of information on protein-protein interactions from the biological literature. Bioinform. (2001) 17:155-161
16. Pustejovsky, J., Castano, J., Zhang, J.: Robust Relational parsing over Biomedical Literature: Extracting Inhibit Relations. Pac. Symp. Biocomput. (2002) 505-516
17. Rindflesch, T. C., Rajan, J. V., Hunter, L.: Extracting Molecular Binding Relationships from Biomedical Text. ANLP. (2000) 188-195
18. Wattarujeekrit, T., Shah, P., Collier, N.: PASBio: predicate-argument structures for event extraction in molecular biology. BMC Bioinformatics. (2004) 5: 155
19. Tapanainen, P., Jarvinen, T.: A non-projective dependency parser. ANLP. (1997) 64-71