

Extraction of Biomedical Events

Roser Morante, Vincent Van Asch, and Walter Daelemans

CLiPS - University of Antwerp

Abstract

In this paper we describe a memory-based machine learning system that extracts biomedical events from texts relying on information from contextual and syntactic features. The main characteristics of the system are that it uses information from dependency syntax and that it integrates classifiers that learn event triggers and event participants jointly. The results show that this system is more efficient than a similar memory-based system that used shallow context information and integrated classifiers in a traditional pipeline architecture.

1 Introduction

In recent years, research on biomedical text mining has seen substantial progress, (Krallinger and Valencia 2005, Ananiadou and McNaught 2006, Krallinger et al. 2008a). The focus on extraction of event frames using machine learning techniques is relatively new, since most research was devoted to named entity recognition. This was due in part to the lack of annotated corpora. Recently, some corpora have been annotated with event level information of different types: PropBank-style frames (Wattarujeekrit et al. 2004, Chou et al. 2006), frame independent roles (Kim et al. 2008, Pyysalo et al. 2007), and specific roles for certain event types (Sasaki et al. 2008). Thanks to these efforts it is possible now to extract biomedical events by applying machine learning techniques.

Most work on biomedical information extraction focuses on extracting relations between biomedical entities within a text. For example, Bundschuh et al. (2008) developed a system to identify relations between genes and diseases from a set of Gene Reference Into Function phrases. Progress in this field has been boosted by the shared tasks on protein-protein interaction extraction in the framework of the Language Learning in Logic Workshop 2005 (Nédellec 2005) and the BioCreative competitions (Krallinger et al. 2008b). Event extraction has emerged to satisfy new information extraction needs. Research on event extraction has benefited from the data and tools made available for the BioNLP Shared Task on Event Extraction 2009 (BioNLP-ST) (Kim et al. 2009), which consisted of finding event triggers and event participants.

In this paper we describe a machine learning system that extracts event triggers and event participants from biomedical texts. The system has been trained and tested on the BioNLP-ST. Although the approach is possible using any classification-based supervised learning method, we chose for Memory-Based Learning (MBL) as learning method. Memory-based language processing

(Daelemans and van den Bosch 2005) is based on the idea that NLP problems can be solved by reuse of solved examples of the problem stored in memory. Given a new problem, the most similar examples are retrieved, and a solution is extrapolated from them.

The originality of the system that we present lies in the fact that event triggers and participants are learned jointly, whereas the machine learning systems that were submitted to the task first learn the event triggers, and then the event participants. It has been shown that jointly learning two tasks can lead to better results than learning the tasks apart in a cascade. Wang et al. (2008) jointly learn Chinese word segmentation, named entity recognition, and part-of-speech tagging, outperforming a pipeline architecture baseline. Finkel and Manning (2009) show that joint learning of parsing and named entity recognition produce mildly improved performance for both tasks. Our goal in this paper is to investigate whether the joint setting is suitable for event extraction.

In Section 2 we briefly describe the data and the task. Section 3 introduces the system architecture. Sections 4 and 5 present the system in detail, and Section 6 the results. Finally, some conclusions are put forward in Section 7.

2 Data and task description

The event extraction system that we present has been trained and evaluated with the data provided by the BioNLP-ST¹, which consisted of extracting bio-molecular events from texts, focusing on events involving proteins and genes. The system was developed after the competition finished. In this task, an *event* is defined as a relation that holds between one or more *entities* that fulfill different roles. There are two types of entities: proteins and biomedical events. Entities can be either *participants* or *arguments* of an event. Participants fulfill the core roles (Theme, Cause) in the event, and arguments (Location, Site) further specify the events. In Sentence (6.1), the proteins *STAT1*, *STAT3*, *STAT4*, *STAT5a*, and *STAT5b* are participants of the biomedical event *phosphorylation*. They fulfill the role *Theme*. *Tyrosine* is an argument of the same event, and it fulfils the role *Site*.

(6.1) IFN-alpha enhanced tyrosine phosphorylation of STAT1, STAT3, STAT4, STAT5a, and STAT5b

For this sentence, the task consists of identifying *phosphorylation* as an event trigger, and extracting the five events listed in Table 6.1.

The event types annotated in the corpora are: Gene Expression, Localization, Phosphorylation, Protein Catabolism, Transcription, Binding, and (Positive, Negative) Regulation. All of them have one *Theme* as participant, except for Binding that can have more than one. Regulations have also a participant *Cause*. Sentence (6.2) contains triggers (*binding*) of Binding events with multiple Themes. The events to be extracted are listed in Table 6.2.

¹Web page: <http://www-tsujii.is.s.u-tokyo.ac.jp/GENIA/SharedTask/>

	Theme	Argument		Theme	Argument
Event 1	STAT1	tyrosine	Event 4	STAT5a	tyrosine
Event 2	STAT3	tyrosine	Event 5	TAT5b	tyrosine
Event 3	STAT4	tyrosine			

Table 6.1: Events to be extracted from Sentence (6.1).

- (6.2) When we analyzed the nature of STAT proteins capable of binding to IL-2Ralpha, pim-1, and IRF-1 GAS elements after cytokine stimulation, we observed IFN-alpha-induced binding of STAT1, STAT3, and STAT4, but not STAT5 to all of these elements.

	Theme(s)	Argument		Theme(s)	Argument
Event 1	STAT4, IRF-1	GAS elements	Event 7	IL-2, Ralpha	GAS elements
Event 2	STAT3, IL-2Ralpha	GAS elements	Event 8	pim-1	GAS elements
Event 3	STAT3, IRF-1	GAS elements	Event 9	STAT1, IRF-1	GAS elements
Event 4	STAT4, pim-1	GAS elements	Event 10	STAT3, pim-1	GAS elements
Event 5	STAT1, IL-2Ralpha	GAS elements	Event 11	IRF-1	GAS elements
Event 6	STAT4, IL-2Ralpha	GAS elements	Event 12	STAT1, pim-1	GAS elements

Table 6.2: Events to be extracted from Sentence (6.2).

Events can be single or nested. A nested event has as argument another event, like in Sentence (6.3), where *effects* triggers a Regulation event, which has as a participant the Gene Expression event *production*.

- (6.3) We have studied the effects of prednisone (PDN) on the production of cytokinase (IL-2, IL-6, TNF-alpha, IL-10).

The training corpus provided for the task consists of 176,146 words and 8,597 events, the development corpus of 33,937 words and 1,809 events, and the test corpus of 57,367 words and 3,182 events. The corpora are annotated with gold standard protein and named entities. The task consists of two subtasks: 1) detecting the triggers of events, that is, the words that express the event, and the triggers of participants and arguments that are not proteins; 2) detecting participants and arguments per event.

The system is evaluated in terms of precision, recall and F1 using the evaluation scripts of the BioNLP-ST. We provide intermediate results of the system based on the development data and we provide the final results on development and test data. As in the official results of the BioNLP-ST, here we will report results under the mode Approximate Span Matching and Approximate Recursive Matching (ASM/ARM) as defined in (Kim et al. 2009). In ASM mode, the requirement of exactly matching the text span of triggers is relaxed. The ARM mode relaxes the requirement for recursive event matching, so that an event can be correct even if

the events it refers to are only partially correct. More details about the task setting, corpora, and evaluation can be found in the webpage of the task and in Kim et al. (2009).

3 System architecture

The architecture of the system is represented in Figure 1. It works in three phases: preprocessing, classification and postprocessing.

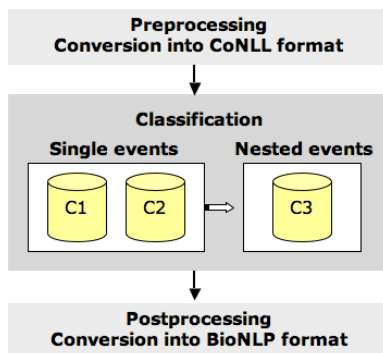


Figure 1: Architecture of the event extraction system.

The system starts by preprocessing the corpora into a learnable format. In order to get information for feature construction for the machine learner, we process the corpora with the GDep dependency parser (Sagae and Tsujii 2007), which outputs for every word the part-of-speech (POS) tag, the lemma, IOB-style chunks, named entities, the syntactic head, and the dependency relation. The data are converted into a column format, following the standard format of the CoNLL Shared Task 2006 (Buchholz and Marsi 2006). Table 6.3 shows a simplified example of a pre-processed sentence. The first column contains the token number in the sentence; the second the word; the third to the sixth contain information provided by the GDEP parser; the seventh column contains the named entities as provided by the BioNLP-ST. The eighth and ninth columns contain as many slots separated by “:” as there are events in the sentence; the eighth column marks with the event type the tokens that are event triggers, and the ninth column marks the tokens that are event participants in the same slot as the event to which they belong. For example, Token 4 “alter” expresses two Regulation events. The first one has as participant Token 9, and the second one Token 11. Triggers are expressed with BI tags (Beginning, Inside) in order to capture multiword triggers.

Rewriting the development corpus into the column format and converting it back into the BioNLP-ST format with gold standard information results in an F1-score of 93.57 %. This score effectively constitutes an upperbound for a machine learner using this data format. The loss in performance can be attributed to the fact

that we do not process intersentential event participants(they amount to 5% of the data), and to the complicated structures of the nested events.

#	WORD	POS	NE	DEP	LABEL	NE TASK	EVENT TRIGGERS	PARTICIPANTS
1	RCC-S	NNS	O	2	SUB	O		
2	did	VBD	O	0	ROOT	O		
3	not	RB	O	2	VMOD	O		
4	alter	VB	O	2	VC	O	B-Reg:B-Reg:	
5	the	DT	O	7	NMOD	O		
6	cytoplasmic	JJ	O	7	NMOD	O		
7	levels	NNS	O	4	OBJ	O		
8	of	IN	O	7	NMOD	O		
9	RelA	NN	B-protein	11	NMOD	B-Protein		Theme:Theme
10	and	CC	O	11	NMOD	O		
11	NF-kappaB1	NN	B-protein	8	PMOD	B-Protein		:Theme:Theme:
12	but	CC	O	2	VMOD	O		
13	did	VBD	O	2	VMOD	O		
14	suppress	VB	O	13	VC	O	:B-NegReg:B-NegReg:	
15	their	PRPS	O	17	NMOD	O		
16	nuclear	JJ	O	17	NMOD	B-Entity		ToLoc:ToLoc
17	localization	NN	O	14	OBJ	O	:B-Loc:B-Loc	:Theme:Theme:
18	and	CC	O	2	VMOD	O		
19	inhibited	VBD	O	2	VMOD	O		
20	the	DT	O	21	NMOD	O		
21	activation	NN	O	19	OBJ	O		
22	of	IN	O	21	NMOD	O		
23	RelA	NN	B-protein	27	NMOD	B-Protein		
24	/	SYM	I-protein	27	NMOD	O		
25	NF-kappaB1	NN	I-protein	27	NMOD	B-Protein		
26	binding	NN	I-protein	27	NMOD	O		
27	complexes	NNS	I-protein	22	PMOD	O		
28	.	.	O	2	P	O		

Table 6.3: Example sentence represented in CoNLL format.

The classification component will be described in Section 4. The postprocessing component consists of three rules that are applied to the output of the classification in order to reconstruct the data into the original BioNLP-ST files. Further details are presented in Section 5.

4 Classification of event triggers and participants

The classification component consists of three memory-based classifiers. Two classifiers are used to find triggers and participants of single events, and one classifier is used to find triggers and participants of nested events, based on the output of the first two classifiers.

The algorithm used is TRIBL as implemented in TiMBL (version 6.1.2) (Daelemans et al. 2007). TRIBL is a hybrid combination of IB1, a k -NN classifier, and IGTREE, a fast decision-tree approximation of k -NN (Daelemans and van den Bosch 2005) that splits the classification of instances into a quick decision-tree traversal based on the first, most informative, features, followed by a slower k -NN classification based on the remaining features. In this case the classifier used the two most informative features for the IGTREE classification. The k -NN classifier is IB1, the usual memory-based algorithm based on the k -nearest neighbor classification rule.

Parameterisation of the classifiers was performed by experimenting with sets of parameters on the development set. TRIBL was parameterised in this case by using gain ratio for feature weighting, overlap as distance metric, 5 nearest neighbors for extrapolation, and normal majority voting for class voting weights. The three

classifiers use the same parameters.

In the next subsections the three classifiers are described and intermediate results are provided.

4.1 Learning single events

Two classifiers are used to find triggers of single events and their participants. Instances represent combinations of tokens that are tagged as Proteins and all the tokens in the sentence with POS verb, noun or adjective, which amount to almost 99% of the events in the training corpus. The three classifiers learn the combined label “EventType:ParticipantType”.

Classifier 1 (C1) processes the instances in which the combined token is an ancestor of the Protein in the dependency tree, and Classifier 2 (C2) processes the rest of the cases. This is motivated by the fact that the predictive power of features is different, as will be explained below. For the sentence in Table 6.3, C1 processes the combinations of protein *RelA* (Token 9) with Tokens 11, 7, 4, and 2, and C2 processes the combination of the same protein with tokens 1, 6, 9, 13, 14, 16, 17, 19, 21, 23, 25, 26, and 27.

We experimented successfully with the features that we list below. Each group of features is tagged with an identifier in bold characters that will be used in the Tables below where we analyze the performance of the classifier per group of features. We mark with (*) the features that are not used by C1.

- Information about protein and combined token: **[Basic]** word, lemma, POS tag, chunk tag, named entity (NE) tag, and dependency label.
- Information about the context of protein and combined token: **[Context]** the same features as for protein and combined token, for a window of three tokens to the right and three to the left in the sequence of tokens, and n-grams of 1, 2, and 3 tokens for the same window of tokens.
- Information about the path in the dependency tree:

[Path] Feature indicating who is the ancestor in the dependency tree (protein, combined token, none)*; boolean feature indicating whether combined token is head of protein; number of steps up from protein; number of steps from common ancestor to combined token* if there is a common ancestor; chains of lemmas, POS, and dependency labels from protein to common ancestor if there is one, or to combined token for C1, and from common ancestor, if there is one, to combined token*;

[Path 2] POS, lemma and dependency label of the tokens that are three steps up from protein and three steps up from the combined token, and from the tokens that are three steps down from the common ancestor in the direction of the protein or from combined token in C1, and three steps down from the common ancestor to the combined token.

- Information about the head and children of protein: [**ProtFam**] Word, lemma, POS, chunk tag NE and dependency label of head, and strings of POS, chunk tags, NE and dependency labels of the children.
- Information about the head and children of combined token: [**EvFam**] the same as for head and children of protein.

Table 6.4 shows the F1 scores of C1 per event type. The number under the event type expresses the frequency of the class.

Features	Event type									
	Bin	GEx	Loc	Pho	PrC	Reg	+Reg	-Reg	Trans	
	195	260	39	35	12	37	127	43	64	
Basic	53.97	77.30	75.32	80.95	84.61	25.53	34.86	25.45	61.65	
+Context	58.40	78.26	74.35	82.92	84.61	27.45	47.34	31.74	60.86	
+Path	61.29	79.41	74.35	80.95	91.66	26.41	50.19	34.92	65.11	
+Path 2	65.94	79.85	80.00	82.35	91.66	20.00	53.84	22.22	64.12	
+ProtFam	66.66	79.77	80.95	83.33	91.66	20.83	54.26	21.87	64.12	
+EvFam	65.39	77.26	79.51	83.33	91.66	26.92	54.26	19.67	61.31	

Table 6.4: F1 results of C1 on the development set in CoNLL format. “Bin”: Binding; “GEx”: Gene Expression; “Loc”: Localization; “Pho”: Phosphorylation; “PrC”: Protein Catabolism; “Reg”: Regulation; “+Reg”: Positive Regulation; “-Reg”: Negative Regulation; “Trans”: Transcription.

As expressed by the “+” character in the first column, features are successively added to the Basic features. We observe that the scores per groups of features are not homogeneous for all event types. It is difficult to find a combination of features that increases the performance for all event types, and it is also difficult to evaluate the scores for the classes that are not frequent. The final system incorporates the version of the classifier that uses the groups of features +ProtFam. The motivation is that these features score the highest for Binding and Positive Regulation, and score only 0.12 lower than the highest for Gene Expression, which are the most frequent event types. These features score also the highest for Localization, Phosphorylation, and Protein Catabolism.

The behavior of features for C1 contrasts with the behavior of features for C2, the results of which are shown in Table 6.5. C2 processes a bigger and more imbalanced data set. In this case, we observe two main characteristics: the Basic features score clearly lower than the highest scoring combination of features, and adding the context features produces lower scores for most event types. When Context features are used, the two most informative features, which TRIBL uses to split the classification, were features about the second and third token to the right of Combined Token, whereas when Context features are not used TRIBL uses the lemma and the word of the Combined Token. Additionally, we also observe that the scores of C2 are much lower than the scores of C1, suggesting that it is more difficult to classify instances in which Protein and Combined Token do not have a dependency relation. The final system incorporates the version of C2 that uses

the groups of features Basic, Path, Path 2, and ProtFam. In general, the most informative features are lemmas of the event context in the sequence of words and in the dependency tree.

Features	Event type								
	Bin	GEx	Loc	Pho	PrC	Reg	+Reg	-Reg	Trans
Basic	20.15	22.01	12.50	36.36	36.36	0.00	12.90	0.00	0.00
+Context	38.09	6.66	0.00	0.00	0.00	13.33	3.84	0.00	0.00
+Path	3.80	6.66	0.00	0.00	0.00	13.79	7.54	0.00	0.00
+Path 2	3.80	6.59	0.00	0.00	0.00	14.81	7.54	0.00	0.00
+ProtFam	3.80	6.66	0.00	0.00	0.00	13.79	7.54	0.00	0.00
+EvFam	3.80	6.66	0.00	0.00	0.00	14.28	7.54	0.00	0.00
Basic	20.15	22.01	12.50	36.36	36.36	0.00	12.90	0.00	0.00
+Path	33.96	29.41	22.22	43.47	50.00	11.76	13.33	0.00	0.00
+Path 2	38.50	32.11	30.00	38.09	53.33	10.52	31.16	0.00	8.00
+ProtFam	41.46	33.56	19.04	36.36	40.00	9.52	25.00	0.00	8.33
+EvFam	37.28	35.21	19.04	34.78	40.00	9.75	23.68	0.00	9.09

Table 6.5: F1 results of C2 on the development set in CoNLL format.

The results of evaluating the system with only C1 and C2 are shown in Table 6.6². Binding and Regulation events score lower, which can be expected because of the possibility of multiple Themes for Binding events and the nested events in Regulations. A positive aspect of these results is that precision is reasonably high.

	Total	Precision	Recall	F1
Binding	248	43.75	28.23	34.31
Gene Expression	356	75.08	63.76	68.96
Localization	53	72.09	58.49	64.58
Phosphorylation	47	65.45	76.60	70.59
Protein Catabolism	21	93.33	66.67	77.78
Transcription	82	66.15	52.44	58.50
Regulation	169	28.00	4.14	7.22
Positive Regulation	617	53.42	12.64	20.45
Negative Regulation	196	26.09	3.06	5.48
TOTAL	1789	61.34	28.62	39.03

Table 6.6: Evaluation of the system with C1 and C2 (ASM/ARM) on development data.

²The total number of events in Table 6.6 is not equal to the sum of the total number of events in Table 6.4 plus the total number of events in Table 6.5 because Tables 6.4 and 6.5 count the events in the CoNLL representation, whereas Table 6.6 counts the events in the BioNLP-ST format. The conversion into CoNLL format is not perfect, as indicated in Section 3.

4.2 Learning nested events

In order to find nested events, we add Classifier 3 (C3). Instances represent combinations of a protein or an event predicted by C1 and C2, and all the tokens in the sentence with POS verb, noun or adjective. Some features used by C3 are different from the ones used by C1 and C2:

- **[Lemmas]** Lemmas of the protein/predicted event and of the combined token.
- **[Basic]** Word, POS tag, chunk tag, named entity (NE) tag, and dependency label of the protein/predicted event and of the combined token.
- **[Context]** of the protein/predicted event and of the combined token. The same features as [Basic], for a window of three tokens to the right and three to the left in the sequence of tokens.
- **[Path]** Same as for C1 and C2.

Table 6.7 shows the results of C3 on development data. In contrast with C1 and C2, the Basic features do not achieve results comparable to the highest scores and the +Path features provoke a clear increase in the scores for all event types, yielding the best combination of features.

Features	Event type									
	Bin	GEx	Loc	Pho	PrC	Reg	+Reg	-Reg	Trans	
	297	344	55	46	21	101	347	121	82	
Lemmas	17.74	33.94	51.61	40.00	63.15	20.28	29.25	6.89	16.07	
+Basic	34.00	38.53	52.42	54.11	61.11	16.26	19.65	10.44	27.82	
+Context	44.44	59.84	46.15	61.01	77.55	18.42	39.80	22.98	48.97	
+Path	56.88	71.84	68.68	73.58	85.71	34.48	53.37	39.19	55.69	

Table 6.7: F1 results of C3 on the development set in CoNLL format.

The most informative features for this classifier are: the feature indicating what is the ancestor; the lemma and word of combined token; the string of lemmas from protein to common ancestor or to combined token in the dependency tree; the full form of protein; the number of steps down from common ancestor to combined token, or from protein if protein is the ancestor; the lemmas of the token to the left and one and two tokens to the right of combined token; the dependency label of combined token.

5 Postprocessing

The Classification phase produces a multicolumn file, as shown in Table 6.8. Some heuristics are needed to rewrite the multicolumn file into the original BioNLP-ST format. We defined three rules: multiple events rule, multiple themes rule, nested events rule.

token	word	event trigger 1	participants 1	event trigger 2	participants 2
1	We	-	-	-	-
2	have	-	-	-	-
3	studied	-	-	-	-
4	the	-	-	-	-
5	effects	Regulation	-	-	-
6	of	-	-	-	-
7	prednisone	-	-	-	-
8	(-	-	-	-
9	PDN	-	-	-	-
10)	-	-	-	-
11	on	-	-	-	-
12	the	-	-	-	-
13	production	-	Theme	Gene_expression	-
14	of	-	-	-	-
15	cytokinase	-	-	-	-
16	(-	-	-	-
17	IL-2	-	-	-	Theme
18	,	-	-	-	-
19	IL-6	-	-	-	Theme
20	,	-	-	-	-
21	TNF-alpha	-	-	-	Theme
22	,	-	-	-	-
23	IL-10	-	-	-	Theme
24)	-	-	-	-

Table 6.8: Output of the Classification component.

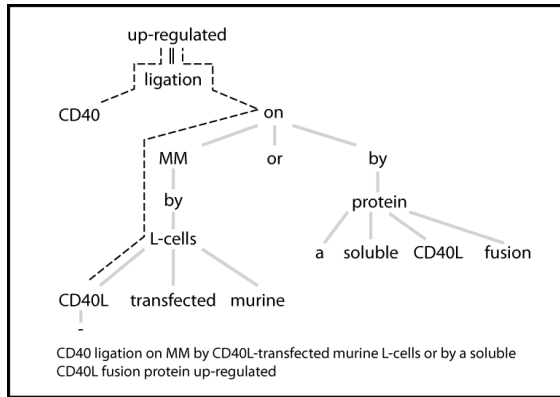
5.1 The multiple events rule

For the sentence in Table 6.8, two events have been predicted. Event 1 expressed by Token 5 is a Regulation event with Event 2 (Token 13) as Theme. Event 2 is a Gene Expression event that has four Themes (Tokens 17, 19, 21, 23). However, the task specifications indicate that a Gene Expression event can have only one Theme. In order to rewrite the output, the system reads in all tokens and the associated predictions containing information about event triggers and participants. One event is created for every predicted trigger and it is assigned a unique identification number. When more than one Theme is predicted for an event, this event is duplicated. As for the example in Table 6.8, the Gene Expression event with four Themes will be split up into four Gene Expression events, each with its own Theme.

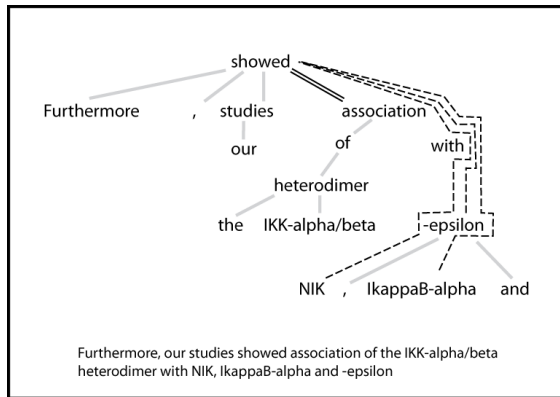
5.2 The multiple themes rule

Binding events can have multiple Themes. By analyzing the data we found that the syntactic information contains clues about the differences between a Binding event with multiple Themes and multiple Binding events with one Theme.

Figure 2 exemplifies cases where a Binding event has multiple Themes. In these cases, the common path to the root of the dependency tree is the same for



(a) Multiple Themes for a Binding event



(b) Multiple Binding events

Figure 2: Multiple Themes for a Binding event versus multiple Binding events

all Themes and for the event trigger. In Figure 2a the common path between the dashed lines (the Themes) is also common with the doubled line (the event trigger). If there are multiple Binding events –all with one Theme– the common path of the Themes is longer than the part that they share with the event trigger, which might indicate that there is a syntactic construction that acts as a coordination. Figure 2b shows that the common path for the dashed lines contains the token “with”. The path to this token is not common with the path of the event trigger. The system uses this path information to disambiguate between multiple Themes for a Binding event and multiple Binding events.

5.3 The nested events rule

Another rule takes care of nested events, like the Regulation event in Table 6.8. For the Gene Expression event in Table 6.8 the first rule created four events from the same event trigger. The rule for nested events will add as many Regulation events as there are new Theme events, in this case Gene Expression events. As a result of the first rule and this rule, the system outputs eight events: four Gene Expression events and four Regulation events, which, in this case, happens to be the same as the gold standard.

6 Results and discussion

The results that we report in this Section have been obtained by training the system on the training corpus and testing it on the test corpus via the web service provided in the web page of the BioNLP Shared Task³. The final results are presented in Table 6.9.

	Total	Precision	Recall	F1
Binding	347	43.46	23.92	30.86
Gene Expression	722	74.12	52.35	61.36
Localization	174	76.74	37.93	50.77
Phosphorylation	135	69.05	64.44	66.67
Protein Catabolism	14	30.00	21.43	25.00
Transcription	137	60.26	34.31	43.72
Regulation	291	33.94	12.71	18.50
Positive Regulation	983	36.59	19.43	25.38
Negative Regulation	379	39.33	15.57	22.31
TOTAL	3182	53.37	29.89	38.32

Table 6.9: Results of the system (ASM/ARM) on the test set.

The scores show that the system does not process Regulation and Binding events at a satisfactory competitive level, like most participating systems⁴. Both, precision and recall are low. Protein Catabolism also gets low scores, but the frequency of these events is extremely low, so the score is not reliable. Precision is acceptable for Gene Expression, Localization, Phosphorylation and Transcription events, and recall for Phosphorylation events. The system reaches an F1 score of 38.32 % in the ASM/ARM mode. The results of this system can be compared to the results of the systems that participated in the BioNLP-ST. There was a large range of variation in the results of participating systems (between 16 and 52% F1

³Web page of the evaluation server: <http://www-tsujii.is.s.u-tokyo.ac.jp/GENIA/SharedTask/eval-test.shtml>.

⁴Results of all the participating systems can be found at <http://www-tsujii.is.s.u-tokyo.ac.jp/GENIA/SharedTask/results/results-master.html>.

score). Compared to the best three systems of the task, the system presented in this paper scores 13.63, 8.34 and 6.3 % lower.

The three best systems used diverse approaches. In Björne et al. (2009), a graph representation and transformation approach is used in which the different steps of the processing (trigger detection and role detection) are graph transformations (adding or removing nodes and edges) achieved using a multi-class SVM. A hand-crafted rule-based postprocessing step patches the output (e.g. removing extra themes for events that can have only one theme). In Buyko et al. (2009), manually curated dictionaries were used as extra information for event trigger identification. This approach starts from dependency parses which are simplified and decorated with conceptual class information. Maximum entropy and graph kernel SVM machine learning algorithms were used for classification (sometimes combined in an ensemble) in different combinations for different types of events. A postprocessing module is used here as well. In Kilicoglu and Bergler (2009), a rule-based system was used, operating on dependency parses and using manually curated dictionaries like UMLS for trigger identification.

Compared to a memory-based system (Morante et al. 2009) that participated in the BioNLP-ST, this system scores 7.75 % F1 higher. The system described in (Morante et al. 2009) solves the classification task in two not-joint learning steps, as most systems do. First, a token-based classification finds event triggers, and then a pairwise classification finds participants. Additionally, that system uses more rules for postprocessing, which are more complex. The difference in precision is 5.67 %, and in recall 7.39 %. The comparison would suggest that the joint approach combines better with memory-based learning, though this conclusion should be further explored, since the classifiers used in both systems do not use the same features. The main difference lies in the fact that the system presented here uses features from the dependency tree, whereas the system in Morante et al. (2009) uses mostly features from the sequential context in the sentence.

It is difficult to gain insight into the decisions made by the machine learner in order to explain the cause of misclassification errors. A low percentage of errors is caused by the lack of intersentential event finding, which amount to 5 % of the cases in the training data. Another percentage of errors is caused by the conversion from BioNLP-ST to CoNLL format (7.43% F1 on development data). Some errors are related to Binding events that have multiple Themes, and to Regulation events with nested structures. Another source of errors are overlapping events, that is, events that are triggered by the same token. Finally, the system has problems with detecting multitoken event triggers like “had only a slight effect”, which is a Positive Regulation event. In sum, apart from the difficulties of the classification tasks, a proportion of errors is caused by the conversion of the data into a learnable format and by the reconstruction of the output of the classification phase into event frames.

7 Conclusions

In this paper we presented a supervised machine learning system that extracts events from biomedical texts according to the definition of the BioNLP Shared Task 2009 (Kim et al. 2009). The main characteristic of the system is that it integrates classifiers that learn event triggers and event participants jointly, avoiding the traditional pipeline architecture prone to error propagation. Additional characteristics are that the system does not make use of external resources like ontologies or dictionaries, and that the rules needed to reformat the data into the original data files are kept to the minimum number making the system adaptable to any domain. The results show that this system is more efficient than a similar memory-based system that used shallow context information and integrated classifiers in a traditional pipeline architecture.

However, the fact that the system scores 13.63 % lower than the best system for the same task suggests that research has to continue in order to reach a more satisfactory performance. Further research will experiment with other algorithms and ensembles because we believe that, given the complexity of the task, taking profit of the positive aspects of different algorithms might increase the performance and help produce results at the level of other well established tasks. Additionally, research should focus on determining what type of domain knowledge would be useful to solve the task and on integrating it in the system.

Acknowledgments

This work was made possible through financial support from the University of Antwerp (GOA project BIOGRAPH).

References

- Ananiadou, S. and J. McNaught (2006), *Text Mining for Biology and Biomedicine*, Artech House Books, London.
- Björne, J., J. Heimonen, F. Ginter, A. Airola, T. Pahikkala, and T. Salakoski (2009), Extracting complex biological events with rich graph-based feature sets, *Proceedings of BioNLP 2009: Shared Task on Event Extraction*.
- Buchholz, S. and E. Marsi (2006), CoNLL-X Shared Task on Multilingual Dependency Parsing, *Proceedings of X CoNLL: Shared Task*, New York.
- Bundschuh, M., M. Dejori, M. Stetter, V. Tresp, and H-P Kriegel (2008), Extraction of semantic biomedical relations from text using conditional random fields, *BMC Bioinformatics* **9**, pp. 207.
- Buyko, E., E. Faessler, J. Wermter, and U. Hahn (2009), Event extraction from trimmed dependency graphs, *Proceedings of BioNLP 2009: Shared Task on Event Extraction*.
- Chou, W.C., R.T.H. Tsai, Y-S. Su, W. Ku, T-Y Sung, and W-L Hsu (2006), A Semi-Automatic Method for Annotating a Biomedical Proposition Bank, *Proceedings of the ACL Workshop on Frontiers in Linguistically Annotated Corpora 2006*.

- Daelemans, W. and A. van den Bosch (2005), *Memory-based language processing*, CUP.
- Daelemans, W., J. Zavrel, K. Van der Sloot, and A. Van den Bosch (2007), TiMBL: Tilburg Memory Based Learner, version 6.1, Reference Guide, *TR 07-07*, ILK, Tilburg, The Netherlands.
- Finkel, J.R. and C.D. Manning (2009), Joint parsing and named entity recognition, *Proceedings of HLT-NAACL 2009*, Boulder, Colorado.
- Kilicoglu, H. and S. Bergler (2009), Syntactic dependency based heuristics for biological event extraction, *Proceedings of BioNLP 2009: Shared Task on Event Extraction*, Boulder, USA.
- Kim, J.D., T. Ohta, and J. Tsujii (2008), Corpus annotation for mining biomedical events from literature, *BMC Bioinformatics* **9**, pp. 10.
- Kim, J.D., T. Ohta, and J. Tsujii (2009), Overview of BioNLP'09 Shared Task on Event Extraction, *Proceedings of BioNLP 2009: Shared Task on Event Extraction*, Boulder, USA.
- Krallinger, M., A. Valencia, and L. Hirschman (2008a), Linking genes to literature: text mining, information extraction, and retrieval applications for biology, *Genome Biology* **9(Suppl 2)**, pp. S8.
- Krallinger, M. and A. Valencia (2005), Text-mining and information-retrieval services for molecular biology, *Genome Biology* **6**, pp. 224.
- Krallinger, M., F. Leitner, C. Rodriguez-Penagos, and A. Valencia (2008b), Overview of the protein–protein interaction annotation extraction task of BioCreative II, *Genome Biology* **9(Suppl 2)**, pp. S4.
- Morante, R., V. Van Asch, and W. Daelemans (2009), A memory-based approach to event extraction in biomedical texts, *Proceedings of BioNLP 2009: Shared Task on Event Extraction*, Boulder, USA.
- Nédellec, C. (2005), Learning Language in Logic – Genic Interaction Extraction Challenge, *Proceedings of LLL 2005*, Bonn, Germany.
- Pyysalo, S., F. Ginter, J. Heimonen, J. Björne, J. Boberg, J. Järvinen, and T. Salakoski (2007), BioInfer: a corpus for IE in the biomedical domain, *BMC Bioinformatics*.
- Sagae, K. and J. Tsujii (2007), Dependency parsing and domain adaptation with LR models and parser ensembles, *Proceedings of CoNLL 2007: Shared Task*, Prague, Czech Republic, pp. 82–94.
- Sasaki, Y., P. Thompson, P. Cotter, J. McNaught, and S. Ananiadou (2008), Event frame extraction based on a gene regulation corpus, *Proceedings of Coling 2008*, Manchester, UK.
- Wang, X., J. Nie, D. Luo, and X. Wu (2008), A Joint Segmenting and Labeling Approach for Chinese Lexical Analysis, *Proceedings of the European conference on Machine Learning and Knowledge Discovery*, Springer Verlag, Berlin, pp. 538–549.
- Wattarujeekrit, T., P.K. Shah, and N. Collier (2004), PASBio: predicate-argument structures for event extraction in molecular biology, *BMC Bioinformatics* **5**, pp. 155.

