Semantic Parsing for Biomedical Event Extraction

Deyu Zhou[1] and Yulan He[2]

[1]School of Computer Science and Engineering, Southeast University,China
[2]Knowledge Media Institute, The Open University, UK

Abstract

We propose a biomedical event extraction system, HVS-BioEvent, which employs the hidden vector state (HVS) model for semantic parsing. Biomedical events extraction needs to deal with complex events consisting of embedded or hierarchical relations among proteins, events, and their textual triggers. In HVS-BioEvent, we further propose novel machine learning approaches for event trigger word identification, and for biomedical events extraction from the HVS parse results. Our proposed system achieves an F-score of 49.57% on the corpus used in the BioNLP'09 shared task, which is only two points lower than the best performing system by UTurku. Nevertheless, HVS-BioEvent outperforms UTurku on the extraction of complex event types. The results suggest that the HVS model with the hierarchical hidden state structure is indeed more suitable for complex event extraction since it can naturally model embedded structural context in sentences.

1 Introduction

In the past few years, there has been a surge of interests in utilizing text mining techniques to provide in-depth bio-related information services. With an increasing number of publications reporting on protein-protein interactions (PPIs), much effort has been made in extracting information from biomedical articles using natural language processing (NLP) techniques. Several shared tasks, such as LLL [7] and BioCreative [4], have been arranged for the BioNLP community to compare different methodologies for biomedical information extraction.

Comparing to LLL and BioCreative which primarily focus on a simple representation of relations of bio-molecules, i.e. protein-protein interaction, the BioNLP'09 Shared Task [5] involves the recognition of bio-molecular events in scientific abstracts, such as gene expression, transcription, protein catabolism, localization and binding, plus (positive or negative) regulation of proteins. The task concerns the detailed behavior of bio-molecules, and can be used to support the development of biomedical-related databases. In the BioNLP'09 shared task evaluation, the system constructed by UTurku [2] achieved an F-score of 51.95% on the core task, the best results among all the participants.

In this paper, we describe a system, called HVS-BioEvent, which employs the hidden vector state model (HVS) to automatically extract biomedical events from biomedical literature. The HVS model has been successfully employed to extract PPIs [9]. However, it is not straightforward to extend the usage of the HVS model for biomedical events extraction. There are two main challenges. First, comparing to the trigger words used for PPIs which are often expressed as single words or at most two words, the trigger words for biomedical event are more complex. For example, controlled at transcriptional and post-transcriptional levels, spanning over 6 words, is considered as the trigger word for the regulation event. In addition, the same word can be the trigger word for different types of biomedical events in different context. Second, biomedical events consist of both simple events and complex events. While simple events are more similar to PPIs which only involve binary or pairwise relations, complex events involve both n-ary ($n > 2$) and nested relations. For example, a regulation event may take another event as its theme or cause which represents a structurally more complex relation. Being able to handle both simple and complex events thus poses a huge challenge to the development of our HVS-BioEvent system.

The rest of the paper is organized as follows. Section 2 presents the overall process of the HVS-BioEvent system, which consists of three steps, trigger words identification, semantic parsing based on the HVS model, and biomedical events extraction from the HVS parse results. Experimental results are discussed in section 3. Finally, section 4 concludes the paper.

2 Biomedical Event Extraction

We perform biomedical event extraction with the following steps. At the beginning, abstracts are retrieved from MEDLINE and split into sentences. Protein names, gene names, trigger words for biomedical events are then identified. After that, each sentence is parsed by the HVS semantic parser. Finally, biomedical events are extracted from the HVS parse results using a hybrid method based on rules and machine learning. All these steps process one sentence at a time. Since 95% of all annotated events are fully annotated within a single sentence, this does not incur a large performance penalty but greatly reduces the size and complexity of the problem. The remainder of the section will discuss each of the steps in details.

2.1 Event Trigger Words Identification

Event trigger words are crucial to biomedical events extraction. In our system, we employ two approaches for event trigger words identification, one is a hybrid approach using both rules and a dictionary, the other treats trigger words identification as a sequence labeling problem and uses a Maximum Entropy Markov Model (MEMM) to detect trigger words.

For the hybrid approach using both rules and a dictionary, firstly, we constructed a trigger dictionary from the original GENIA event corpus [6] by extracting the annotated trigger words. These trigger words were subsequently lemmatized and stemmed. However, the wide variety of potential lexicalized triggers for an event means that lots of triggers lack discriminative power relative to individual event types. For example, in certain context, through is the trigger word for the binding event type and are is the trigger word for localization. Such words are too common and cause potential ambiguities and therefore lead to many false positive events extracted. We could perform disambiguation by counting the co-occurrence of a event trigger and a particular event type from the training data and discard those event triggers whose co-occurrence counts are lower than certain threshold for that event type. After this filtering stage, still, there might be cases where one trigger might representing multiple event types, we thus define a set of rules to further process the trigger words matched from the constructed dictionary.

In the second approach, we treat trigger words identification as a sequence labeling problem and train a first-order MEMM model [8] from the BioNLP'09 shared task training data. As in typical named entity recognition tasks, the training data are converted into BIO format where 'B' refers to the word which is the beginning word of an event trigger, 'I' indicates the rest of the words (if the trigger contains more than one words) and 'O' refers to the other words which are not event triggers. The features used in the MEMM model was extracted from the surface string and the part-of-speech information of the words corresponding to (or adjacent to) the target BIO tags.

2.2 Semantic Parsing using the HVS Model

The Hidden Vector State (HVS) model [3] is a discrete Hidden Markov Model (HMM) in which each HMM state represents the state of a push-down automaton with a finite stack size. State transitions are factored into separate stack pop and push operations constrained to give a tractable search space. The sequence of HVS stack states corresponding to the given parse tree is illustrated in Figure 1. The result is a model which is complex enough to capture hierarchical structure but which can be trained automatically from only lightly annotated data.

In the HVS-based semantic parser, conventional grammar rules are replaced by three probability tables. Let each state at time t be denoted by a vector of D_t semantic concept labels (tags) $c_t = [c_t[1], c_t[2], ..c_t[D_t]]$ where $c_t[1]$ is the preterminal concept label and $c_t[D_t]$ is the root concept label (SS in Figure 3). Given a word sequence W, concept vector sequence $\mathbf{C}$ and a sequence of stack pop operations N, the joint probability of $P(W, \mathbf{C}, N)$ can be decomposed as

$$P(W, \mathbf{C}, N) = \prod_{t=1}^{T} P(n_t|c_{t-1}) P(c_t[1]|c_t[2 \cdots D_t]) P(w_t|c_t) \quad (1)$$

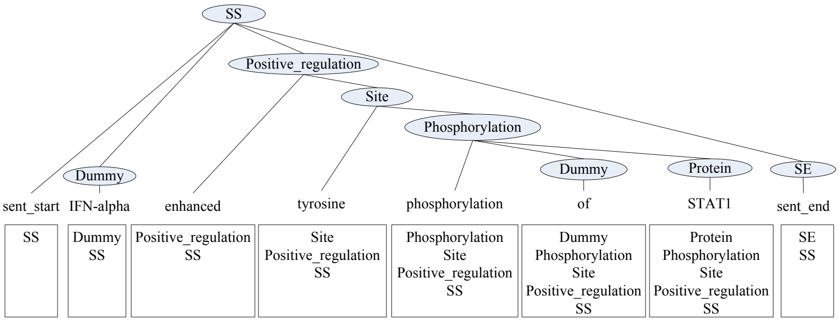
SS
Positive_regulation
Site
Phosphorylation
Dummy
Dummy
Protein
SE
sent_start
IFN-alpha
enhanced
tyrosine
phosphorylation
of
STAT1
sent_end
SS
Dummy SS
Positive_regulation SS
Site Positive_regulation SS
Phosphorylation Site Positive_regulation SS
Dummy Phosphorylation Site Positive_regulation SS
Protein Phosphorylation Site Positive_regulation SS
SE SS

Figure 1: Example of a parse tree and its vector state equivalent.

where n_t is the vector stack shift operation and takes values in the range $0, \cdots, D_{t-1}$, and $c_t[1] = c_{w_t}$ is the new pre-terminal semantic label assigned to word w_t at word position t.

Thus, the HVS model consists of three types of probabilistic move, each move being determined by a discrete probability table: (1) popping semantic labels off the stack - $P(n|c)$; (2) pushing a pre-terminal semantic label onto the stack - $P(c[1]|c[2 \cdots D])$; (3) generating the next word - $P(w|c)$. Each of these tables are estimated in training using an EM algorithm and then used to compute parse trees at run-time using Viterbi decoding. In training, each word string W is marked with the set of semantic concepts C that it contains. For example, the sentence IFN-alpha enhanced tyrosine phosphorylation of STAT1 contains the semantic concept/value pairs as shown in Figure 1. Its corresponding abstract semantic annotation is:
`Positive_regulation(Site(Phosphorylation(protein)))`
where brackets denote the hierarchical relations among semantic concepts[1]. For each word w_k of a training sentence W, EM training uses the forward-backward algorithm to compute the probability of the model being in stack state c when w_k is processed. Without any constraints, the set of possible stack states would be intractably large. However, in the HVS model this problem can be avoided by pruning out all states which are inconsistent with the semantic concepts associated with W. The details of how this is done are given in [3].

For the sentences in the BioNLP'09 shared task, only event information is provided. However, the abstract semantic annotation as shown above is required for training the HVS model. We propose Algorithm 1 to automatically convert the annotated event information into the abstract semantic annotations. An example of how the abstract annotations are generated is given as follows.
Sentence: According to current models the inhibitory capacity of I(kappa)B(alpha) would be mediated through the retention of Rel/NF-kappaB proteins in the cytosol.

Corresponding Events:	E1	Negative_regulation: inhibitory_capacity	Theme: I(kappa)B(alpha)
	E2	Positive_regulation: mediated	Theme: E1

Candidate annotation generation (Steps 1-4 of Algorithm 1):
Negative_regulation(Protein)   Negative_regulation(Protein(Positive_regulation))
Abstract annotation pruning (Steps 5-14 of Algorithm 1):
Negative_regulation(Protein(Positive_regulation))

2.3 Biomedical Events Extraction From HVS Parse Results

Based on HVS parse results, it seems straightforward to extract the event information. However, after detailed investigation, we found that sentences having the same semantic tags might contain different events information. For example, the two sentences shown in Table 1 have the same semantic parsing results but with different event information.

This problem can be solved by classification. For the semantic tags which can represent multiple event information, we considered each event information as a class and employed hidden Markov support vector machines (HM-SVMs) [1] for disambiguation among possible events. The features used in HM-SVMs are extracted from surface strings and part-of-speech information of the words corresponding to (or adjacent to) trigger words.

[1] We omit SS and SE here which denote sentence start and end.

Algorithm 1 Abstract semantic annotation generation.

```
Input: A sentence $W =< w_1, w_2, \cdots, w_n >$, and its event information $Ev =< e_1, e_2, \cdots, e_m >$
Output: Abstract semantic annotation $A$
for each event $e_i$ =<Event_type:Trigger_words Theme:Protein_name ...> do
    Sort the Trigger_words, Protein_name, and other arguments based on their positions in $W$ and get a sorted list $t_1, t_2, ..., t_k$
    Generate an annotation as $t_1(t_2(..t_k))$, add it into the annotation list $A$
end for
for each annotation $a_i \in A$ do
    if $a_i$ contains another event then
        Replace the event with its corresponding annotation $a_m$
    end if
end for
for each annotation $a_i \in A$ do
    if $a_i$ is a subset of another annotation in $A$ then
        Remove $a_i$ from the annotation list $A$
    end if
end for
Reorder annotations in $A$ based on their positions in $W$
```

Sentence	We concluded that CTCF expression and activity is controlled at transcriptional and post-transcriptional levels	CONCLUSION: IL-5 synthesis by human helper T cells is regulated at the transcriptional level
Parse results	SS+Protein(CTCF) SS+Protein+Gene_Expression(expression) SS+Protein+Gene_Expression+Regulation(controlled...levels)	SS+Protein(IL-5) SS+Protein+Gene_Expression(synthesis) SS+Protein+Gene_Expression+Regulation(regulated)
Events	E1 Gene_expression:expression Theme: CTCF E2 Regulation: controlled...levels Theme: E1 E3 Regulation: controlled...levels Theme: CTCF	E1 Gene_expression: synthesis Theme: IL-5 E2 Regulation: regulated Theme: E1

Table 1: An example of the same semantic parse results denoting different event information

3 Results and Discussion

Experiments have been conducted on the training data of the BioNLP'09 shared task which consists of 800 abstracts. After cleaning up the sentences which do not contain biomedical events information, 2893 sentences were kept. We split the 2893 sentences randomly into the training set and the test set at the ratio of 9:1 and conducted the experiments ten times with different training and test data each round.

Method	*Recall* (%)	*Precision* (%)	*F-score* (%)
Trigger Word Identification			
Dictionary+Rules	46.31	53.34	49.57
MEMM	45.43	40.91	42.99
Event Extraction from HVS Parse Results			
No classification	43.57	52.85	47.77
With Classification	46.31	53.34	49.57

Table 2: Experimental results based on 10 fold cross-validation.

Table 2 shows the performance evaluated using the approximate recursive matching method adopted from the BioNLP'09 share task evaluation mode. To evaluate the performance impact of trigger word identification, we also report the overall performance of the system using the two approaches we proposed, dictionary+rules and MEMM. The results show that the hybrid approach combining a trigger dictionary and rules gives better performance than MEMM which only achieved a F-score around 43%. For biomedical event extraction from HVS parse results, employing the classification method presented in Section 2.3 improves the overall performance from 47.77% to 49.57%.

The best performance that HVS-BioEvent achieved is an F-score of 49.57%, which is only two points lower than UTurku, the best performing system in the BioNLP'09 share task. It should be noted that our results are based on 10-fold cross validation on the BioNLP'09 shared task training data only since we don't have the access to the BioNLP'09 test set while the results generated by UTurku were evaluated on the BioNLP'09 test set. Although a direct comparison is not possible, we could still speculate that

Simple Events			*Complex Events*		
Event Class	*HVS-BioEvent*	*UTurku*	*Event Class*	*HVS-BioEvent*	*UTurku*
localization	61.40	**61.65**	binding	**49.90**	44.41
gene expression	72.44	**73.90**	regulation	**36.57**	30.52
transcription	**68.30**	50.23	negative regulation	**40.61**	38.99
protein catabolism	**70.27**	52.17			
phosphorylation	56.52	**77.58**			

Table 3: Per-class performance comparison in F-score (%) between HVS-BioEvent and UTurku.

HVS-BioEvent is comparable to the best performing system in the BioNLP'09 shared task.

The results on the five event types involving only a single theme argument are shown in Table 3 as *Simple Events*. For the complex events such as "binding", "regulation" and "negative regulation" events, the results are shown in Table 3 as *Complex Events*. We notice that HVS-BioEvent outperforms UTurku on the extraction of the complex event types, with the performance gain ranging between 2% and 7%. The results suggest that the HVS model with the hierarchical hidden state structure is indeed more suitable for complex event extraction since it could naturally model embedded structural context in sentences.

4 Conclusions

In this paper, we have presented HVS-BioEvent which uses the HVS model to automatically extract biomedical events from text. The system is able to offer comparable performance compared with the best performing system in the BioNLP'09 shared task. Moreover, it outperforms the existing systems on complex events extraction which shows the ability of the HVS model in capturing embedded and hierarchical relations among named entities. In future work we will explore incorporating arbitrary lexical features into the HVS model training in order to further improve the extraction accuracy.

References

[1] Y. Altun, I. Tsochantaridis, and T. Hofmann. Hidden markov support vector machines. In *International Conference in Machine Learning*, pages 3–10, 2003.

[2] Jari Bjorne, Juho Heimonen, Filip Ginter, Antti Airola, Tapio Pahikkla, and Tapio Salakoski. Extracting complex biological events with rich graph-based feature sets. In *Proceedings of the Workshop on BioNLP*, pages 10–18, 2009.

[3] Yulan He and Steve Young. Semantic processing using the hidden vector state model. *Computer Speech and Language*, 19(1):85–106, 2005.

[4] Lynette Hirschman, Alexander Yeh, Christian Blaschke, and Alfonso Valencia. Overview of biocreative: critical assessment of information extraction for biology. *BMC Bioinformatics*, 2005.

[5] Jin-Dong Kim, Tomoko Ohta, Sampo Pyysalo, Yoshinobu Kano, and Jun'ichi Tsujii. Overview of bionlp'09 shared task on event extraction. In *Proceedings of the Workshop on BioNLP*, pages 1–9, 2009.

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

[7] Claire Nédellec. Learning Language in Logic - Genic Interaction Extraction Challenge. In *Learning Language in Logic workshop (LLL05)*, pages 31–37, 2005.

[8] Nam Nguyen and Yunsong Guo. Comparisons of sequence labeling algorithms and extensions. In *Proceedings of the ICML*, pages 681–688, 2007.

[9] Deyu Zhou, Yulan He, and Chee Keong Kwoh. Extracting protein-protein interactions from medline using the hidden vector state model. *International Journal of Bioinformatics Research and Applications*, 4(1):64–80, 2008.