Biomedical Events Extraction using the Hidden Vector State Model

Deyu Zhou[a,*], Yulan He[b]

[a]*School of Computer Science and Engineering, Southeast University, Nanjing, Jiangsu Province, China, 210093*
[b]*Knowledge Media Institute, The Open University, Walton Hall, Milton Keynes, United Kingdom, MK7 6AA*

Abstract

Objective: Biomedical events extraction concerns about extracting events describing changes on the state of bio-molecules from literature. Comparing to the protein-protein interactions (PPIs) extraction task which often only involves the extraction of binary relations between two proteins, biomedical events extraction is much harder since it needs to deal with complex events consisting of embedded or hierarchical relations among proteins, events, and their textual triggers. In this paper, we propose an information extraction system based on the hidden vector state (HVS) model, called HVS-BioEvent, for biomedical events extraction, and investigate its capability in extracting complex events.

Methods and Material: HVS has been previously employed for the extractions of PPIs. In HVS-BioEvent, we propose an automated way to generate abstract annotations for HVS training and further propose novel machine learning approaches for event trigger word identification, and for biomedical events extraction from the HVS parse results.

Results Our proposed system achieves an F-score of 49.57% on the corpus used in the BioNLP'09 shared task, which is only 2.38% lower than the best performing system by UTurku in the BioNLP'09 share task. Nevertheless, HVS-BioEvent outperforms UTurku's system on complex events extraction with

*Corresponding author. Tel:+86-255209876; Fax: +86-255209876.
Email addresses: d.zhou@seu.edu.cn (Deyu Zhou), y.he@open.ac.uk (Yulan He)

36.57% vs 30.52% being achieved for extracting regulation events, and 40.61% vs 38.99% for negative regulation events.

Conclusions 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.

Keywords: Hidden vector state model, biomedical events extraction, abstract annotations, semantic parsing.

---

1. Introduction

In the past few years, there have 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 [1] and BioCreative [2], have been arranged for the BioNLP community to compare different methodologies for biomedical information extraction. In general, existing PPI extraction approaches can be roughly categorized into three types, machine learning methods [3], approaches based on pattern matching [4] and those employing parsing techniques [5].

Comparing to protein-protein interactions which often only involves binary relations between two proteins, bio-molecular events describing changes on the state of bio-molecules are more complex. For example, "Spc97p interacts with Spc98 and Tub4 in the two-hybrid system" describes two PPIs, Spc97p interacts with Spc98 and Spc97p interacts with Tub4. However, "...inhibiting tyrosine phosphorylation of STAT6..." describes two bio-molecular events, one is the phosphorylation event, the other is the complex or embedded negative regulation event which is signaled by the word inhibiting and takes the first phosphorylation event as its argument. In a typical biomedical event annotation, we can represent these two events as:

E1 (Event Type:Phosphorylation, Theme:STAT6, ToLoc:tyrosine)

E2 (Event Type: Negative_regulation:inhibiting Theme:E1)

Bio-molecular events extraction aims to extract such event information from biomedical literature and reformats these extracted information in structures as represented by the two annotations presented above. By extracting detailed behaviors of bio-molecules, bio-molecular event extraction can be used to support the development of biomedical-related databases.

The BioNLP'09 Shared Task [6] is the recent one focusing on 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. In the shared task evaluation, the system constructed by Jari *et al.* [7] achieved an F-score of 51.95% on the core task, the best results among all the participants. The best F-score result obtained is still relatively low, mainly attributed to the following two main reasons, one is the large variety of the event trigger words and the other is the complexity of the sentences to be dealt with.

To tackle the complexity of the sentences, we constructed a system, called HVS-BioEvent, which uses the hidden vector state model (HVS) to automatically extract biomedical events from biomedical literature. The HVS model [8] 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. It is complex enough to capture hierarchical structure but which can be trained automatically from only lightly annotated data. The HVS model has been successfully employed to extract PPIs [5]. 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.

We summarize our contributions below. First, we have proposed an automated way to generate abstract annotations from the BioNLP'09 shared task data and successfully deployed the HVS model for biomedical events extraction. Second, we have proposed two novel machine learning approaches, one for event trigger word identification, and another 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 2.38% lower than that of UTurku's system, the best performing system in this task. Nevertheless, HVS-BioEvent outperforms UTurku's system on complex events extraction with 36.57% vs 30.52% being achieved for extracting regulation events, and 40.61% vs 38.99% for negative regulation events.

The rest of the paper is organized as follows. Section 2 discusses the related work on biomedical events extraction, followed by a brief description of the BioNLP'09 shared task. Section 3 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 4. Finally, section 5 concludes the paper.

2. Related Work

2.1. Work on Protein-Protein Interactions Extraction

Approaches proposed to extract PPIs can be roughly categorized into three types, machine learning methods, rule-based methods and those employing parsing techniques.

Rule-based methods generally achieve better performance compared to other categories. For example, Ono *et al.* [9] manually defined some linguistic patterns which were then augmented with additional constraints based on word forms and syntactic categories to generate better matching precision. It achieved high performance with a recall rate of 85% and a precision rate of 84% for *Saccharomyces cerevisiae (yeast)* and *Escherichia coli*. However, these methods are not feasible in practical applications as they require heavy manual efforts to define patterns when shifting to other domains.

Machine learning approaches to the PPIs extraction task typically cast it as a classification problem where a sentence containing a pair of proteins is classified as implying interaction of the pair or not. Features used for classifier training are normally syntactic and lexical patterns derived from dependency relations between individual words in sentences which are revealed automatically by syntactic parsers. Various kernels have been proposed to calculate similarity between syntactic structures, including subsequence kernel [10], tree kernels [11], shortest path kernel [12], graph kernel [13], or a combination of them [14]. Under this kind of problem setting, one sentence in the dataset yields C_n^2 distinct instances, where n is the number of proteins in the sentence and each instance represents a pairwise combination of proteins.

Approaches employing parsing techniques make use of semantic parsing models. One example is the hidden vector state (HVS) model [8] which can map sentences to their semantic meaning representations without the use of expensive tree-bank style training data. The model has been employed successfully for extracting PPIs [5]. To further improve the performance of the HVS-based system, other techniques such as semi-supervised learning [15], discriminative training [16] and the hybrid training framework [17] have been proposed.

2.2. Work on Biomedical Events Extraction

The prevailing approaches to relation extraction has focused on extracting pairwise or binary relations. McDonald *et al.* [18] has attempted to extract n-ary (for $n > 2$) relations by factoring higher-order relations into a set of

binary relations and using a classifier to extract binary relations. Entities graph is then created and higher-order relations are constructed by finding maximal cliques. Still, there has been very little work in extracting complex relations, in particular, nested relations, such biomedical events information.

Recently, two corpora annotated with complex, nested and typed event relations have been introduced, the BioInfer [19] and GENIA Event [20]. The two corpora aim to capture the diversity of biological relations. The GENIA Event corpus was used in the BioNLP'09 Shared Task which aims to extract nested bio-molecular events from research abstracts, where an event may have variable number of arguments and may contain other events as arguments. Most participants to the Shared Task either reduced the task to binary classification problem or used heuristics to combine manual rules and statistics. Among 24 submissions, the best result with an F-score of 51.95% was obtained by Bjorne *et al.* [7] who essentially transformed complex relation extraction into binary classification. A classifier (such as SVMs) needs to be trained for every relation type seen in the training data, which thus hinders its scalability. Ozgur and Radev [21] also trained a separate SVM classifier for different event types, but only achieved an overall F-score around 40%. Farzaneh *et al.* [22] identified the event participants using a rule-based system which relies on a relative distance between candidate entities and the trigger in the associated parse tree. The overall F-score is around 30%. Jörg *et al.* [23] presented an approach based on a deep parser using the Link Grammar. It gave an overall F-score of 29.6%. More recently, Poon and Vanderwende [24] proposed a joint approach for bio-event extraction based on Markov logic but still trailed the previously reported best approach [7] by two points on the BioNLP'09 Shared Task test set.

2.3. The BioNLP'09 Shared Task

The BioNLP'09 Shared Task concerns the recognition of bio-molecular events that appear in biomedical literature. The shared task consists of three subtasks, *Core event extraction*, *Event enrichment*, and *Negation and speculation recognition*. Table 1 illustrates with three example sentences where their events

information corresponds to the three subtasks. *Core event extraction*, as shown in the first row of Table 1, includes trigger detection (Expression), event typing (Gene_expression:Expression), primary argument recognition (IkappaBalpha) and finally fill into the frame (E1 event_type:event trigger Theme:primary argument). For *Event enrichment*, the secondary arguments are found and added into the event frame as ToLoc: nuclear as shown in the second row of Table 1. For *Negation and speculation recognition*, negations and speculations of events need to be identified and formatted as M1 Negation/Speculation E1 where E1 denotes the event information recognized in the *Core event extraction* and *Event enrichment subtasks*.

The organizers provide human-curated reference material for the training and evaluation of the participating systems. For training, a data set based on the publicly available GENIA corpus is provided in a stand-off format.

3. System Overview

The overall architecture of the system is shown in Figure 1. 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 combining 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. An example of using HVS-BioEvent for biomedical event extraction is illustrated in Figure 2. For the sentence "All agents tested induced expression of Hsp60 6 hr after application.", the event trigger words "induced", "expression" are replaced separately with their corresponding event types "positive_regulation" and "gene_expression" at the event trigger words identification step as shown in Figure 2(a). At the semantic parsing step, the HVS model generates the parsing result of the sentence

as presented in Figure 2(b) where symbols preceding the parentheses such as "SS+POSITIVE_REGULATION" are the semantic tags. Finally, the event extraction step extracts the event information as shown in Figure 2(c). The remainder of the section will discuss each of the steps in details.

3.1. Event Trigger Words Identification

Event trigger words are crucial to biomedical events extraction. Approaches for biomedical term identification (such as protein name, gene name) can also be used for event trigger words detection. They typically fall into three categories, dictionary based, rule based and machine learning based. In our system, we explored two approaches for event trigger words identification, one is a hybrid approach combing a dictionary and rules, the other treats trigger words identification as a sequence labeling problem and uses a Maximum Entropy Markov Model (MEMM) to detect trigger words.

The hybrid approach first constructs a trigger word dictionary from the original GENIA event corpus [20]. The corpus consists of 1,000 Medline abstracts with 36,114 events being annotated. We extracted annotated event trigger words together with their corresponding event types. For example, (stimulates, positive_regulation) denotes that the trigger word "stimulates" triggers the event "positive_regulation". Then these trigger words were lemmatized and stemmed. Thus, the above example would be changed to (stimulate, positive_regulation). After that, duplicate entries were removed and the remaining entries were sorted according to their occurrence frequencies in the corpus. Table 2 lists the top 10 most frequency trigger word/event type entries.

By examine the sorted list, we found that lots of trigger words are too common and lack the discriminative power relative to individual event types. For example, in certain context, through is the trigger word for the binding event type and therefore is the trigger word for positive_regulation. Using such words for even type identification would cause potential ambiguities and therefore might lead to many false positive events extracted. However, such common words typically occur much less frequent in denoting event types compared

to their overall occurrences in the corpus. For example, through occurs 311 times in the corpus, but only appears once to denote the binding event. Hence, we could calculate the ratio between the occurrence frequency of each trigger word in denoting an event type and its total occurrences in the corpus. Those trigger words with their ratios below certain threshold are discarded. In our experiments here, we empirically set the threshold to 0.05. After the processing, 3771 entries were kept. Table 3 gives some example entries whose ratios are below the threshold.

After the filtering stage, there might be cases where one trigger might represent multiple event types. For example, underlie in some context denotes the regulation event type, while in other context denotes the correlation event type. Thus, it is important to disambiguate which event type it refers to. We proposed a rule-based approach for event type disambiguation. First, for each ambiguous event trigger word, we collected the sentences containing such a word in the GENIA event corpus. Then we selected words occurring before or after the trigger word within some predefined window size and converted them to word features. A decision tree was built for each trigger word using these word features. Finally, rules were extracted automatically from these decision trees for event type disambiguation. Below is an example rule generated for the trigger word underlie:

IF the word following "underlie" is a gene or protein related term, the word "underlie" is **not** an event trigger;
ELSE IF the word following "underlie" is another biomedical event, the word "underlie" triggers the event type **Regulation**;
ELSE the word "underlie" triggers the event type **Correlation**;
END.

In the second approach, we treat trigger words identification as a sequence labeling problem and train a first-order Maximum entropy Markov model (MEMM) [25] on the BioNLP'09 shared task training data. Maximum entropy Markov models are based on the concept of a probabilistic finite state model such as the Hidden Markov model (HMM). However, instead of generating observations

as in HMM, MEMM consider observation sequences to be conditioned upon. Given a finite set of states S and a finite output alphabet X, MEMM only need to define a single set of S separately trained distributions $P(s'|s, x)$. The distributions represent the probability of moving from state s to s' on observation x. Thus, the conditional distribution over a label sequences $\mathbf{y}$ given an observation sequence $\mathbf{x}$ is:

$$p(\mathbf{y}|\mathbf{x}) = p(y_1 = s_1|x_1) \prod_{t=2}^{n} p(y_t = s_t|y_{t-1} = s_{t-1}, x_t) \quad (1)$$

To treat trigger words identification as a sequence labeling problem, three labels 'B', 'I', and 'O 'are introduced 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. Then the training data were converted into BIO format. The features used in the MEMM model were extracted from the surface string and the part-of-speech information of the words corresponding to (or adjacent to) the target BIO tags. Given a word sequence (a sentence), MEMM output a tag sequence where each word is tagged as one of the 'B', 'I', or 'O' tags. It can then be easily identified the trigger word(s) from the BIO tag sequence.

3.2. Semantic Parsing using the HVS Model

The Hidden Vector State (HVS) model [8] 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. This is illustrated in Figure 3 which shows a sequence of the HVS stack states corresponding to the given parse tree. State transitions are factored into separate stack *pop* and *push* operations constrained to give a tractable search space. The result is a model which is complex enough to capture hierarchical structures but which can be trained automatically from only lightly annotated data.

The HVS model computes a hierarchical parse tree for each word string W, and then extracts semantic concepts C from this tree. Each semantic concept

consists of a name-value pair where the name is a dotted list of primitive semantic concept labels. For example, the top part of Figure 3 shows a typical semantic parse tree and the semantic concepts extracted from this parse would be in Equation 2

```
Positive_regulation = enhanced
Positive_regulation.Site = tyrosine
Positive_regulation.Site.Phosphorylation = phosphorylation
Positive_regulation.Site.Phosphorylation.Protein = STAT1
```
(2)

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

$$P(W, \mathbf{C}, N) = \prod_{t=1}^{T} P(n_t|c_{t-1})P(c_t[1]|c_t[2 \cdots D_t])P(w_t|c_t) \tag{3}$$

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. D_{t-1} denotes the number of semantic concept labels in the vector at word position $t-1$.

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 - $P(w|c)$.

In training, each word string W is marked with the set of semantic concepts C that it contains. For example, if the sentence shown in Figure 3 was in the training set, then it would be marked with the four semantic concepts given in Equation 2. The abstraction semantic annotation for the sentence is

```
SS(Positive_regulation(Site(Phosphorylation(protein))) SE)
```
(4)

where SS and SE denotes sentence start and end and brackets denote the hierarchical relations among semantic concepts. 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. Maximum Likelihood Estimation (MLE) is used for estimating the probabilities using the following re-estimation formulae:

$$P^*(n|c') = \frac{\sum_t P(n_t{=}n, \mathbf{c}_{t-1}{=}c'|W,\lambda^k)}{\sum_t P(\mathbf{c}_{t-1}{=}c', W|\lambda^k)} \tag{5}$$

$$P^*(c[1]|c[2..D]) = \frac{\sum_t P(\mathbf{c}_t, W|\lambda^k)}{\sum_t P(c_t[2..D]{=}c[2..D]|W,\lambda^k)} \tag{6}$$

$$P^*(w|\mathbf{c}) = \frac{\sum_t P(\mathbf{c}_t{=}\mathbf{c}, w_t{=}w|\lambda^k)}{\sum_t P(\mathbf{c}_t{=}\mathbf{c}, W|\lambda^k)} \tag{7}$$

These probabilities are then used to generate parse results at run-time using Viterbi decoding. The time complexity of parsing based on the HVS model is $O(TQ^D)$, where T is the length of the sequence, D is the maximum depth of stack (vector state), and Q is the max number of semantic tags (node labels) at each level of the stack.

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 [8].

For the sentences in the BioNLP'09 shared task, only event information is provided. However, the abstract semantic annotation as in Equation 4 is required for training the HVS model. We proposed Algorithm 1 to automatically convert the annotated event information into the abstract semantic annotations. The time complexity of Algorithm 1 is $O(m^2)$, where m is the number of event information in the sentence W.

An example of abstract semantic annotation generation is shown below:

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

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

Algorithm 1 Abstract semantic annotation generation.

Input: The sentence $W =< w_1, w_2, \cdots, w_n >$, and its corresponding event information $Ev =< e_1, e_2, \cdots, e_m >$, $e_i =$<Event_type:Trigger_words Theme:Protein_name ...>

Output: Abstract semantic annotation A

1: Set $A = \emptyset$
2: **for** $i = 1$ to m **do**
3: Sort the trigger_words, protein_name, and other argument words in event information e_i based on their position in the sentence W and get the sorted list $t_1, t_2, ..., t_k$
4: Set $A[i] = t_1(t_2(..t_k))$, where t_j is the j^{th} words in the sorted list
5: **end for**
6: **for** $i = 1$ to m **do**
7: **if** $A[i]$ contains another event, e.g. E1 **then**
8: Replace the event with its corresponding annotation $A[l]$
9: **end if**
10: **end for**
11: **for** i=1 to m **do**
12: **for** j=i+1 to m **do**
13: **if** $A[i]$ is a subset of $A[j]$ **then**
14: Set $A[i]$ = Null
15: **end if**
16: **end for**
17: **end for**
18: Output the annotations in A and reorder them based on their positions in the sentence W

Candidate annotation generation (steps 1-4 of Algorithm 1):

Negative_regulation(Protein) Negative_regulation(Protein(Positive_regulation))

Abstract semantic annotation (steps 5-14 of Algorithm 1):

SS(Negative_regulation(Protein(Positive_regulation)) SE)

3.3. Biomedical Events Extraction From the HVS Parse Results

Based on the 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 4 have the same semantic parse results but contain different event information.

We analyzed HVS semantic parse results and found that three types of semantic tags need to be disambiguated as shown in Table 5. Consider the event information shown in the right column of Table 5 as classes, the disambiguation problem can be converted to the classification problem. For each of the class, we trained a SVM classifier which takes a sentence as the input and outputs whether it represents the event information that the class corresponds to. Given the semantic tag which needs to be disambiguated, w_i is the word corresponding to the semantic tag and p_i is the part-of-speech (POS) tag of w_i. The features used in SVM training are w_i and its five preceding and five subsequent words, plus p_i and its five preceding and five subsequent POS tags.

Incorporating the disambiguation process mentioned above, the whole procedure of event extraction from HVS parse results is described in Algorithm 2. For each semantic tag, first check whether it requires disambiguation. If so, classification will be invoked. For example, Protein+Gene_expression+Regulation, requires disambiguation as it belongs to one of the ambiguous semantic tag types listed in Table 5. Then check whether the semantic tag ends with an event trigger (e.g. Protein+Localization). If this is the case, search backward to find the theme of the event (since every event should have a theme) and add the event information. Otherwise, check whether the semantic tag ends with a protein or an entity (e.g. Binding+Protein). If so, search backward to find the corresponding trigger word (word with the semantic tag containing Binding) and add the event information. Based on the approach described above, the biomedical events can be extracted as shown in Figure 2(c).

Algorithm 2 Biomedical event extraction from HVS parse results.

Input: The sentence $W =< w_1, w_2, \cdots, w_n >$, and its corresponding semantic tag sequence $S =< s_1, s_2, \cdots, s_n >$. The semantic tag-event list MT in which one semantic tag may represent multiple event information.

Output: Event list $E =< e_1, e_2, \cdots, e_m >$, where $e_i =$<Event_Type:Trigger words, Theme:protein name,...>

```
for each word w_i do
  Compare the semantic tag length l(s_i) of s_i with s_{i-1}.
  if l(s_i) > l(s_{i-1}) then
    if s_i is in the semantic tag-event list MT then
      Perform classification based the S and W
      Search backwards s_{i-1}, ..., s_1 for theme, trigger word, add event information into E
    else if the last tag in s_i is a trigger word then
      Search backwards s_{i-1}, ..., s_1 for theme, add event information into E
    else if the last tag in s_i is a protein or a entity then
      Search backwards s_{i-1}, ..., s_1 for trigger word, add event information into E
    end if
  end if
end for
```

4. 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. The average parsing speed on IBM Linux server equipped with 3.00Ghz processor and 4 GB RAM was 0.14s per sentence. The

average speed for generating the abstract annotation on the server was 4s per 1000 sentences.

Table 6 shows the performance evaluated using the approximate recursive matching method adopted from the BioNLP'09 share task evaluation mode. We also report the overall performance of the system using the two different trigger words identification approaches proposed, dictionary+rules and MEMM. The results show that the hybrid approach combining a dictionary and rules gives better performance than MEMM which only achieved an F-score around 43%. For biomedical event extraction from HVS parse results, employing the classification method presented in Section 3.3 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 2.38% lower than UTurku's system, 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's system were evaluated on the BioNLP'09 test set. Although a direct comparison is not possible, we could still speculate that 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 7 as *Simple Events*. For the complex events such as binding, regulation and negative regulation events, the results are shown in Table 7 as *Complex Events*. It can be observed that HVS-BioEvent achieved F-scores in the range of 57-73% for simple events extraction and 37-50% for complex events extraction. This is not surprising since complex events contain structurally more complex or nested relations and thus it is much more difficult for our system to extract compared to those simple events which only contain pairwise or binary relations.

To investigate our system's ability in handling complex events, we compare the performance of our system with the UTurku's system. Figure 4 shows the comparison on recall, precision and F-score. It can be seen that HVS-BioEvent

outperforms UTurku's system 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.

Based on our knowledge, there is only one system participated in the BioNLP'09 shared task which is modified from AkanePPI, a public available protein-protein interaction extraction system [26]. AkanePPI has previously achieved state-of-the-art performance on all existing public PPI corpora. By adding new types of name entities to represent the events, the modified AkanePPI for event extraction only achieved an F-score of 42.6%. More deliberated design on event trigger identification and the use of HVS for semantic parsing may explain our superior performance compared to the modified AkanePPI.

We also conducted error analysis by analyzing the parse results of 150 randomly selected sentences from the test data set. The errors are classified into three categories as shown in Table 8 together with the total number of sentences falling into each category. We also gave an example sentence for each category, with its extracted events and the gold standard. The three categories of errors are semantic parsing errors, trigger words identification errors and event extraction errors. (1) Semantic parsing errors constitute the major portion of all errors. We found that the current semantic parsing method causes approximately 60% of the total errors. This partially derives from the fact that some complex hierarchical structures still can not be handled correctly by our method. (2) Errors caused by the trigger words identification procedure accounts for nearly 15% of all the failures. (3) Event extraction procedure caused about 25% errors.

5. Conclusions

In this paper, we have presented HVS-BioEvent which uses the HVS model to automatically extract information on 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. Our results may provide a useful supplement to manually created resources in established public databases. In future work we will explore incorporating arbitrary lexical features into the HVS model training in order to further improve the extraction accuracy.

Acknowledgements

We would like to thank the anonymous reviewers for their valuable comments.

References

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

[2] Lynette Hirschman, Alexander Yeh, Christian Blaschke, and Alfonso Valencia. Overview of biocreative: critical assessment of information extraction for biology. *BMC Bioinformatics*, 6(Suppl 1):S1, 2005.

[3] I. Donaldson, J. Martin, B. de Bruijn, and C. Wolting. Prebind and textomy–mining the biomedical literature for protein-protein interactions using a support vector machine. *BMC Bioinformatics*, 4(11), 2003.

[4] Minlie Huang, Xiaoyan Zhu, and Yu Hao. Discovering patterns to extract protein-protein interactions from full text. *Bioinformatics*, 20(18):3604–3612, 2004.

[5] 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 (IJBRA)*, 4(1):64–80, 2008.

[6] 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, Morristown, NJ, USA, 2009. Association for Computational Linguistics.

[7] 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, Morristown, NJ, USA, 2009. Association for Computational Linguistics.

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

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

[10] Razvan Bunescu and Raymond Mooney. Subsequence kernels for relation extraction. In Y. Weiss, B. Schölkopf, and J. Platt, editors, *Advances in Neural Information Processing Systems 18*, pages 171–178, Cambridge, MA, 2006. MIT Press.

[11] Alessandro Moschitti. Making tree kernels practical for natural language learning. In *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics*, pages 113–120, 2006.

[12] Razvan C. Bunescu and Raymond J. Mooney. A shortest path dependency kernel for relation extraction. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 724–731, Morristown, NJ, USA, 2005. Association for Computational Linguistics.

[13] Antti Airola, Sampo Pyysalo, Jari Björne, Tapio Pahikkala, Filip Ginter, and Tapio Salakoski. A graph kernel for protein-protein interaction extraction. In *Proceedings of the Workshop on Current Trends in Biomedical*

Natural Language Processing, pages 1–9, Morristown, NJ, USA, 2008. Association for Computational Linguistics.

[14] Makoto Miwa, Rune Sare, Yusuke Miyao, Tomoko Ohta, and Jun'ichi Tsujii. Combining multiple layers of syntactic information for protein protein interaction extraction. In *Proceedings of the Third International Symposium on Semantic Mining in Biomedicine (SMBM 2008)*, pages 101–108, 2008.

[15] Deyu Zhou, Yulan He, and Chee Keong Kwoh. Semi-Supervised Learning of the Hidden Vector State Model for Extracting Protein-Protein Interactions. *Artificial Intelligence in Medicine*, 41:209–222, 2007.

[16] Deyu Zhou and Yulan He. Discriminative Training of the Hidden Vector State Model for Semantic Parsing. *IEEE Transaction on Knowledge and Data Engineering*, 21:66–77, 2009.

[17] Deyu Zhou and Yulan He. A Hybrid Generative/Discriminative Framework to Train a Semantic Parser from an Un-annotated Corpus. In *Proceedings of 22nd International Conference on Computational Linguistics (COLING2008)*, pages 1113–1120, Manchester, UK, 2008.

[18] McDonald Ryan, Pereira Fernando, Kulick Seth, Winters Scott, Jin Yang, and White Pete. Simple algorithms for complex relation extraction with applications to biomedical ie. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 491–498, Morristown, NJ, USA, 2005. Association for Computational Linguistics.

[19] Sampo Pyysalo, Filip Ginter, Juho Heimonen, Jouni Järvinen Jari Björne, Jorma Boberg, and Tapio Salakoski. BioInfer: a corpus for information extraction in the biomedical domain. *BMC Bioinformatics*, 8(50), 2007.

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

[21] Arzucan Ozgur and Dragomir R. Radev. Supervised classification for extracting biomedical events. In *Proceedings of the Workshop on BioNLP*, pages 111–114, Morristown, NJ, USA, 2009. Association for Computational Linguistics.

[22] Sarafraz Farzaneh, Eales James, Mohammadi Reza, Dickerson Jonathan, Robertson David, and Nenadic Goran. Biomedical event detection using rules, conditional random fields and parse tree distances. In *Proceedings of the Workshop on BioNLP*, pages 115–118, Morristown, NJ, USA, 2009. Association for Computational Linguistics.

[23] Jörg Hakenberg, Illés Solt, Domonkos Tikk, Luis Tari, Astrid Rheinländer, Quang Long Ngyuen, Graciela Gonzalez, and Ulf Leser. Molecular event extraction from link grammar parse trees. In *Proceedings of the Workshop on BioNLP*, pages 86–94, Morristown, NJ, USA, 2009. Association for Computational Linguistics.

[24] Hoifung Poon and Lucy Vanderwende. Joint inference for knowledge extraction from biomedical literature. In *Human Language Technologies: The 11th Annual Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL)*, pages 813–821, Los Angeles, US, 2010.

[25] Nam Nguyen and Yunsong Guo. Comparisons of sequence labeling algorithms and extensions. In *Proceedings of the 24th international conference on Machine learning*, pages 681–688, New York, NY, USA, 2007. ACM.

[26] Rune Satre amd Makoto Miwa, Kazuhiro Yoshida, and Junchi Tsujii. From protein-protein interaction to molecular event extraction. In *Proceedings of the Workshop on BioNLP*, pages 103–106, Morristown, NJ, USA, 2009. Association for Computational Linguistics.

6. Tables

Subtask	*Sentence*	*Events*
Core event extraction	***Expression of IkappaBalpha*** in the nucleus of human peripheral blood T lymphocytes.	**E1 Gene_expression**: Expression **Theme**: IkappaBalpha
Event enrichment	We demonstrate the ***nuclear localization of I(kappa)B(alpha)*** in PBL by different techniques: Western blot, indirect immunofluorescence and electron microscopy.	**E1 Localization**: localization **Theme**: I(kappa)B(alpha) **ToLoc**: nuclear
Negation and speculation recognition	This ***failure to degrade IkappaBalpha*** may underlie both the observed decrease in NFkappaB induction and the IL-2 receptor expression in TNF-treated T cells during aging.	**E1 Protein_catabolism**: degrade **Theme**: IkappaBalpha **M1 Negation** E1

Table 1: Examples of the three subtasks of the BioNLP'09 shared task.

Trigger word	Event type	Word occ. in denoting the event type
express	gene_expression	1748
induce	positive_regulation	1601
active	positive_regulation	1403
inhibit	negative_regulation	864
bind	binding	823
regulate	regulation	602
transcribe	transcription	527
mediate	positive_regulation	424
activate	physiological_process	424
differentiate	cell_differentiation	292

Table 2: The 10 most frequent trigger words in denoting event types in the GENIA-event corpus.

Trigger word	Event type	Word occ. in denoting the event type	Word occ. in the whole corpus	Ratio
being	localization	1	30	0.03
but	correlation	1	531	0.002
by	correlation	1	2564	0.0004
do	positive_regulation	1	52	0.02
after	positive_regulation	2	186	0.01
due	correlation	2	61	0.03

Table 3: Examples of the removed pairs whose ratios are below the threshold.

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 4: An example of the same semantic parse results denoting different event information.

7. Figure captions

7.1. Figure 1. The main components of the system.

7.2. Figure 2. An example of biomedical events extraction using HVS-BioEvent.

7.3. Figure 3. Example of a parse tree and its vector state equivalent.

7.4. Figure 4. Performance comparison between HVS-BioEvent and UTurku's system on complex events extraction.

Semantic tags	*Possible event information represented*
Trigger1+Protein+Trigger2	<E1 Trigger1 Protein> <E2 Trigger2 E1>
	<E1 Trigger2 Protein> <E2 Trigger1 E1>
Trigger1+Trigger2+Protein	<E1 Trigger2 Protein> <E2 Trigger1 E1>
	<E1 Trigger2 Protein> <E2 Trigger1 E1> <E1 Trigger1 Protein>
Protein+Trigger1+Trigger2	<E1 Trigger1 Protein> <E2 Trigger2 E1>
	<E1 Trigger1 Protein> <E2 Trigger2 E1> <E3 Trigger2 Protein>

Table 5: The semantic tag-event list in which a semantic tag denotes multiple event information.

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 6: Experimental results based on 10 fold cross-validation.

Event Class	*Recall* (%)	*Precision* (%)	*F-score* (%)
Simple Events			
localization	56.22	67.63	61.40
gene expression	70.96	73.98	72.44
transcription	64.93	72.05	68.30
protein catabolism	65.00	76.47	70.27
phosphorylation	51.66	62.40	56.52
Complex Events			
binding	44.39	56.96	49.90
regulation	33.73	39.94	36.57
negative regulation	38.24	43.29	40.61

Table 7: Per-class performance in terms of recall, precision, and F-score.

Result Category	*N0. of Sentences*	*Example of the sentence, extracted events, and golden events*
Events identified correctly, but with wrong event information generated	26	Sentence: Targeted mutational analysis demonstrated that a tandem NF-kappa B/Rel binding motif is critical for the gamma 3 ECS responsiveness to both CD40L and IL-4, while a STAT-6-binding site is additionally required for IL-4 inducibility. Extracted: E0 POSITIVE_REGULATION: inducibility Theme: IL-4 E1 POSITIVE_REGULATION: inducibility Theme: STAT-6 E3 POSITIVE_REGULATION: inducibility Theme: CD40L Golden: E1 Positive_regulation: inducibility Theme: IL-4
Events identified partially, without wrong event information generated	40	Sentence: The gene expression of interferon (IFN)-inducible protein 10 (IP-10) (a CXC chemokine) was markedly augmented by the IFNgamma treatment in PMA- or RA-differentiated U937 cells, but only marginally in undifferentiated or VitD3-treated cells. Extracted: E0 GENE_EXPRESSION: gene_expression Theme: IP-10 Golden: E3 Gene_expression: gene_expression Theme: IP-10 E4 Positive_regulation: augmented Theme: E3 Cause:IFNgamma
Events identified partially, with wrong event information generated	37	Sentence: In this study, the influence of the sequences located between -3134 and -2987 on the transcriptional activity of the proIL-1beta gene in LPS-stimulated Raw 264.7 cells was examined in detail Extracted: E1 TRANSCRIPTION: transcriptional activity Theme: proIL-1beta E2 REGULATION: influence Theme: proIL-1beta Golden: E1 Regulation: influence Theme: E2 E2 Transcription: transcriptional activity Theme: proIL-1beta

Table 8: Error analysis from the sample data.