Extracting Protein-Protein Interactions from MEDLINE using the Hidden Vector State Model

Deyu Zhou*, Yulan He and Chee Keong Kwoh

School of Computer Engineering, Nanyang Technological University, Nanyang Avenue, Singapore 639798
E-mail: {zhou0063, asylhe, asckkwoh}@ntu.edu.sg
*Corresponding author

Abstract: Protein-protein interactions referring to the associations of protein molecules are crucial for many biological functions. A major challenge in text mining for biomedicine is automatically extracting protein-protein interactions from the vast amount of biomedical literature since most knowledge about them still hides in biomedical publications. We have constructed an information extraction system based on a semantic parser employing the hidden vector state (HVS) model for protein-protein interactions. Unlike other hierarchical parsing models which require fully annotated treebank data for training, the HVS model can be trained using only lightly annotated data whilst simultaneously retaining sufficient ability to capture the hierarchical structure needed to robustly extract semantics in task domain. When applied it in extracting protein-protein interactions from biomedical literature, we found that it performed better than other established statistical methods and achieved 61.5% in F-score with balanced recall and precision values. Moreover, the statistical nature of the pure data-driven HVS model makes it intrinsically robust and it can be easily adapted to other domains, which is rarely mentioned and possessed by other rule-based approaches.

Keywords: Information extraction, Hidden Vector State Model, protein-protein interactions

Reference to this paper should be made as follows: Deyu Zhou, Yulan He and Chee Keong Kwoh. (2008) 'Extracting Protein-Protein Interactions from MEDLINE using the Hidden Vector State Model', *Int. J. Bioinformatics Research and Applications*, Vol. 4, No. 1, pp.64–80.

Biographical notes: Deyu Zhou currently is a Ph.D candidate in school of computer engineering at Nanyang Technological University. His interests are statistical methods for mining knowledge from texts and biomedical data mining. He received B.S. degree in mathematics in 2000 and M.E. degree in computer science from Nanjing University in 2003.
Yulan He is an Assistant Professor in the School of Computer Engineering at Nanyang Technological University (NTU), Singapore. She obtained her PhD degree in 2004 from Cambridge University, UK, where

she worked in the area of spoken language understanding. Since moving to NTU in 2004, her research interests have been broadened to biomedical literature mining, knowledge integration from heterogeneous data sources, and ontology learning from text. She has published in the areas of spoken dialogue systems, bioinformatics, data and text mining, and information retrieval.
Chee Keong Kwoh received his Ph.D. degree in The Department of Computing at Imperial College, University of London in 1995 and his doctoral thesis was in Probabilistic Reasoning. He is currently working in the School of Computing and hold a joint-appointment in the School of Chemical and Biomedical Engineering, Nanyang Technological University.

1 Introduction

Protein-protein interactions referring to the associations of protein molecules are crucial for many biological functions. Understanding protein functions and how they interact with each other gives biologists a deep insight into understanding of living cells as complex machines, disease process and provides targets for effective drug designs. Although many databases, such as BIND [1], IntAct [2] and STRING [3], have been built to store protein-protein interaction information, constructing such databases is time-consuming and needs immense amount of manual efforts to ensure the correctness of data. To date, vast quantity of knowledge about protein-protein interactions still hides in biomedical literature. As a result, automatically extracting these information from biomedical text holds the promise of easily discovering large amounts of biological knowledge in computer-accessible form.

At the earlier stage of this field, statistical methods [4,5] were employed to search abstracts or sentences which may describe protein-protein interactions based on the co-occurrence of protein names. Following that, other approaches [6, 7] focused on detecting proteins pairs and determining the relations between them based on some probability scores. Obviously, these approaches can not achieve satisfiable performance because they ignore sentence structures which play an important role in expressing protein-protein interactions.

Since then, more and more complicated approaches have been proposed. They can be roughly classified into two categories: those based on pattern matching and those employing parsing techniques. Approaches using pattern matching [8–10] rely on a set of predefined or automatically generated patterns to extract protein-protein interactions. For example, [8] manually defined some patterns which were then augmented with additional restrictions 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.* [9] introduced a probability score for each predefined rule based on its reliability. Interaction events were assigned scores depending on their matched patterns and the distances between protein names. They also considered negative sentences. However, these methods are not feasible in practical applications as they require heavy manual efforts to define patterns when shifting to another domain. Parsing based methods employ either shallow or deep parsing. Shallow parsers [11, 12] break sentences into none-overlapping chunks. Local dependencies are extracted among chunks without reconstructing the structure of an entire sentence. The precision and recall rates of these approaches published so far range from 50% to 80% and from 30% to 80%, respectively. Systems based on deep parsing [13, 14] deal with the structure of an entire sentence and therefore are potentially more accurate. [13] defined a grammar for biomedical domain and used a full parser to extract interaction events. Another full parsing based approach uses the context-free grammar (CFG) to extract protein interaction information with a recall rate of 63.9% and a precision rate of 70.2% [14].

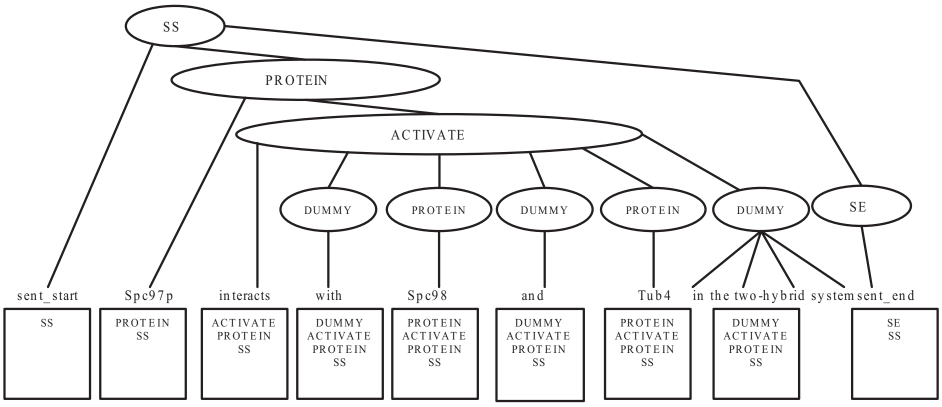
In this paper, we propose an approach based on the hidden vector state (HVS) model to automatically extract protein-protein interactions from biomedical literature. The HVS model has been successfully applied to discover semantic information in spoken utterances [15]. However, it is not straightforward to extend the usage of the HVS model to the biomedical literature domain. One major reason is that spoken utterances are normally simple and short. Unlike written documents, there are normally no complex syntactic structures in spoken utterances. It therefore poses a challenge on how to effectively and efficiently extract semantic information from much more complicated written documents. This paper explores the performance of our approach based on the HVS model for protein-protein interactions extraction [16].

Compared to the previously published approaches, our method has the potential to stand out in several points. Firstly, the HVS model can be easily adapted to other domains by adding a small set of adaption training data. Secondly, by employing the preprocessing method such as sentence simplification and the postprocessing method including relation extraction from the 5-best parsing results, the HVS model gives fairly stable performance on complex sentences. This shows that the ability of our method on handling complex sentence structures is almost the same as that on handling simple sentence structures, which is rarely possessed by the rule-based approaches. In addition, the performance of our approach based on the HVS model is so far the best among all the statistical approaches employing semantic parsing. It is also comparable to the performance of the full parsing approach employing CFG with a recall of 63.9% and a precision of 70.2% [14], although, in general, it is difficult to compare our method with other existing approaches directly, because there is neither an accurate task definition on processing the MEDLINE abstracts nor a benchmark dataset for extracting protein-protein interactions [a].

The rest of the paper is organized as follows: In the next section, we will briefly describe the HVS model and how it can be used to extract protein-protein interactions from biomedical literature. In section 3, we present the overall structure of the extraction system and its components. Experimental results are discussed in section 4. Adaptation methods and results are presented in section 5. Finally, section 6 concludes the paper and gives future directions.

[a] BioCreAtIvE challenge [17] began in 2004 and provided two common evaluation tasks to assess the state of the art for text mining applied to biological problems. Currently, extraction of protein-protein interactions from text is targeted as a main task in BioCreAtIvE II 2006 and evaluation corpora are still not released to the public.

Figure 1 An example of a simplified parse tree and its vector state equivalent.



2 The Hidden Vector State Model

In linguistics, semantic analysis is defined as the process of relating syntactic structures to their language-independent meanings. Given a semantic parse tree for a sentence as illustrated in the upper part of Figure 1, the semantic information relating to each word in the sentence is fully described by the semantic concept or tag ranging from the pre-terminal node to the root node. If storing these semantic information as a label for each word, semantic parsing can be formulated as a sequence labelling problem. Let W denote a word sequence $\langle w_1, w_2, \cdots, w_n \rangle$, the semantic parsing task is to predict a label sequence $S = \langle s_1, s_2, \cdots, s_n \rangle$.

Existing statistical approaches to this problem include sliding-window methods [18], hidden Markov models [19], maximum entropy Markov models [20], conditional random fields [21], graph transformer networks [22] etc [for a review, see [23]].

While the aforementioned approaches require fully annotated training corpora for model parameter estimation in general, we propose a hidden vector state (HVS) model [15] which only needs abstract semantic annotations serving as constraints to limit the forward-backward search during the Expectation Maximization (EM) training. The HVS model 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 1 which shows the sequence of HVS stack states corresponding to the given parse tree. State transitions are factored into a stack shift followed by a push of one or more new preterminal semantic concepts relating to the next input word. If such operations are unrestricted, the state space will grow exponentially and the same computational tractability issues of hierarchical HMMs are incurred. By limiting the maximum stack depth and only allowing one new preterminal semantic concept to be pushed onto the stack for each new input word, the state space can be reduced to a manageable size. The result is a model which is complex enough to capture hierarchical structure but which can be trained automatically from only lightly annotated data.

The HVS model computes a hierarchical parse tree for each word sequence 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 semantic concepts extracted from the parse tree illustrated in the upper part of Figure 1 is shown in equation 1

```
PROTEIN=Spc97
PROTEIN.ACTIVATE=interacts
PROTEIN.ACTIVATE.PROTEIN=Spc98
PROTEIN.ACTIVATE.PROTEIN=Tub4
```
(1)

In the HVS-based semantic parser, conventional grammar rules are replaced by three probability tables. Given a word sequence W, a 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

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

where T is the length of the word sequence W, n_t is the vector stack shift operation, $\mathbf{c}_t$ denotes the vector state at word position t, which consists of D_t semantic concept labels (tags), i.e. $\mathbf{c}_t = [c_t[1], c_t[2], ..., c_t[D_t]]$, $c_t[1] = c_{w_t}$ is the new pre-terminal semantic label assigned to word w_t at word position t and $c_t[D_t]$ is the root concept label (SS in Figure 1).

Thus, the HVS model consists of three types of probabilistic move, each move being determined by a discrete probability table:

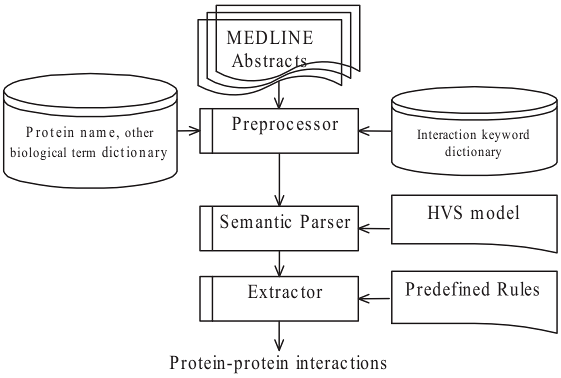
1. popping semantic labels off the stack - $P(n|\mathbf{c})$;
2. pushing a pre-terminal semantic label onto the stack - $P(c[1]|c[2 \cdots D])$;
3. generating the next word - $P(w|\mathbf{c})$.

Each of these probability 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 sequence W is marked with the set of semantic concepts C that it contains. For example, if the sentence shown in Figure 1 was in the training set, then it would be marked with the four semantic concepts given in equation 1. For each word w_k of each 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 [15].

3 System Overview

The overall architecture of the extraction system is shown in Figure 2. It works as follows. At the beginning, abstracts (or full papers) are retrieved from MEDLINE and split into sentences. Protein names and other biological terms are then

Figure 2 System architecture.


MEDLINE Abstracts
Protein name, other biological term dictionary
Preprocessor
Interaction keyword dictionary
Semantic Parser
HVS model
Extractor
Predefined Rules
Protein-protein interactions

identified based on a pre-constructed biological term dictionary. After that, each sentence is parsed by the HVS semantic parser. Finally, protein-protein interactions are extracted from the tagged sentences using a set of manually-defined simple rules. An example of the procedure is illustrated in Figure 3. The details of each step are described below.

1. *Preprocessing - identification of protein names, other biological terms and interaction keywords, simplification of sentences.*

To extract protein-protein interactions from literature, protein names need to be identified firstly, which still remains as a challenging problem. In our system, protein names and other biological terms such as "adenovirus", "NK cells" are identified based on a manually constructed dictionary of biological terms. In addition, a category/keyword dictionary for identifying terms describing interactions has also been built based on [14]. All identified biological terms and interaction keywords are then replaced with their respective category labels as can be seen in Figure 3 (b). By doing so, the vocabulary size of the training corpus can be reduced and the data sparseness problem would be alleviated.

We believe that some types of words, such as articles, adjectives, do not contribute to the expression of protein-protein interactions. POS tagging is employed to parse sentences and these types of words are removed. To avoid removing some adjective words such as "inhibitory" which may indicate protein-protein interaction, words whose etyma can be found in the keyword dictionary are kept.

2. *Parsing sentences using the HVS model.*

Sentences which contain at least two distinct proteins identified by step 1 are then parsed with the HVS model. Before doing so, the HVS model needs to be trained using a lightly annotated training corpus. An annotation example is shown below.

Sentence: CUL-1 was found to interact with SKR-1, SKR-2, SKR-3, SKR-7, SKR-8 and SKR-10 in yeast two-hybrid system.

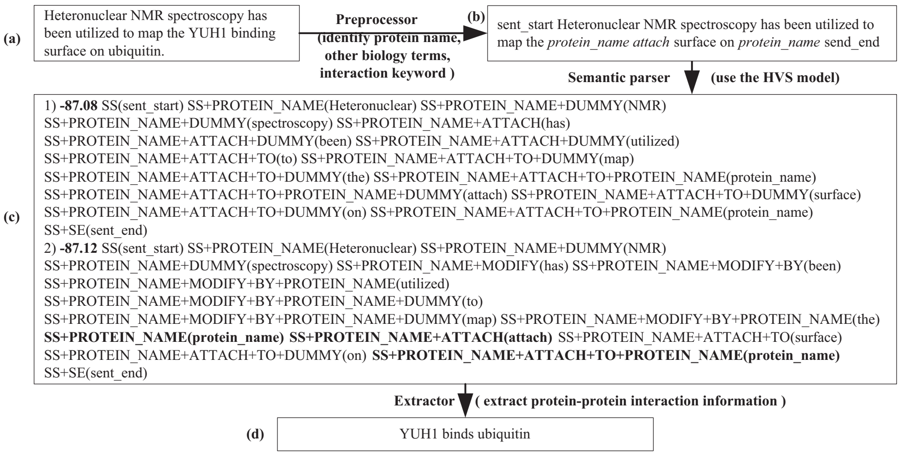
Annotation: PROTEIN_NAME(ACTIVATE(PROTEIN_NAME))

We suspected that prepositions play an important role in expressing embedding semantic relationships, therefore we provided another set of annotations which include the preposition information as shown below:

PROTEIN_NAME(ACTIVATE(WITH(PROTEIN_NAME)))

It can be seen that unlike fully-annotated treebank data, no explicit semantic

Figure 3 An example of a procedure for information extraction using the HVS model.


(a)
Preprocessor
(identify protein name, other biology terms, interaction keyword)
(b)
Semantic parser
(use the HVS model)
(c)
Extractor
(extract protein-protein interaction information)
(d)

Heteronuclear NMR spectroscopy has been utilized to map the YUH1 binding surface on ubiquitin.

sent_start Heteronuclear NMR spectroscopy has been utilized to map the *protein_name attach* surface on *protein_name* send_end

```
1) -87.08 SS(sent_start) SS+PROTEIN_NAME(Heteronuclear) SS+PROTEIN_NAME+DUMMY(NMR)
SS+PROTEIN_NAME+DUMMY(spectroscopy) SS+PROTEIN_NAME+ATTACH(has)
SS+PROTEIN_NAME+ATTACH+DUMMY(been) SS+PROTEIN_NAME+ATTACH+DUMMY(utilized)
SS+PROTEIN_NAME+ATTACH+TO(to) SS+PROTEIN_NAME+ATTACH+TO+DUMMY(map)
SS+PROTEIN_NAME+ATTACH+TO+DUMMY(the) SS+PROTEIN_NAME+ATTACH+TO+PROTEIN_NAME(protein_name)
SS+PROTEIN_NAME+ATTACH+TO+PROTEIN_NAME+DUMMY(attach) SS+PROTEIN_NAME+ATTACH+TO+DUMMY(surface)
SS+PROTEIN_NAME+ATTACH+TO+DUMMY(on) SS+PROTEIN_NAME+ATTACH+TO+PROTEIN_NAME(protein_name)
SS+SE(sent_end)
2) -87.12 SS(sent_start) SS+PROTEIN_NAME(Heteronuclear) SS+PROTEIN_NAME+DUMMY(NMR)
SS+PROTEIN_NAME+DUMMY(spectroscopy) SS+PROTEIN_NAME+MODIFY(has) SS+PROTEIN_NAME+MODIFY+BY(been)
SS+PROTEIN_NAME+MODIFY+BY+PROTEIN_NAME(utilized)
SS+PROTEIN_NAME+MODIFY+BY+PROTEIN_NAME+DUMMY(to)
SS+PROTEIN_NAME+MODIFY+BY+PROTEIN_NAME+DUMMY(map) SS+PROTEIN_NAME+MODIFY+BY+PROTEIN_NAME(the)
SS+PROTEIN_NAME(protein_name) SS+PROTEIN_NAME+ATTACH(attach) SS+PROTEIN_NAME+ATTACH+TO(surface)
SS+PROTEIN_NAME+ATTACH+TO+DUMMY(on) SS+PROTEIN_NAME+ATTACH+TO+PROTEIN_NAME(protein_name)
SS+SE(sent_end)
```

YUH1 binds ubiquitin

tag/word pairs are given. Only the abstract annotations are provided to guide the EM training of the HVS model.

3. *Extracting protein-protein interactions.*
Instead of employing the best parsing result, protein-protein interactions are extracted based on the 5-best parsing results for each sentence (top 2 parsing result examples are shown in Figure 3 (c)). The extracting process follows the rules below:

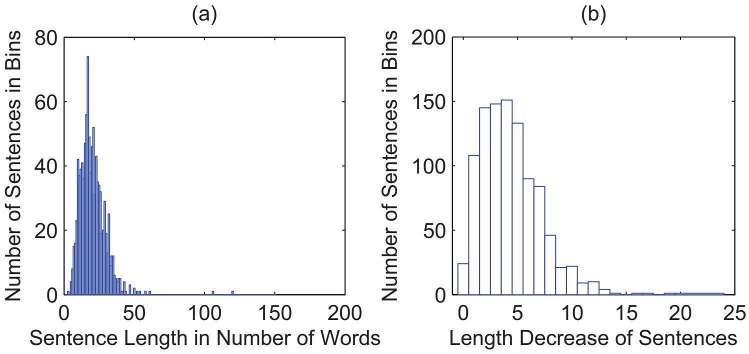
- Ignore the semantic tag if its preterminal tag is DUMMY;

- If an interaction keyword such as "activate", "attach" etc is tagged with "DUMMY" in the best parsing result of a sentence, then check the second best parsing result and so on until this interaction keyword is tagged with its corresponding category label. If such a parsing result can be found, then extract the protein-protein interactions from this parsing output. Otherwise, the best parsing result will still be used. Figure 3 (c, d) illustrates the application of this rule where the protein-protein interaction information is extracted from the second best parsing result, instead of the best one.

- If a semantic tag with the form SS+PROTEIN_NAME+REL+PRO TEIN_NAME or SS+REL+PROTEIN_NAME+PROTEIN_ NAME can be found in the parsing result, where REL can be any of the category names describing the interactions such as "activate", "inhibit" etc, then check whether the corresponding word is in fact a protein name. If so, search backwards or forward for the interaction keyword and the other protein name. Otherwise, ignore this semantic tag.

Based on the rules described above, protein-protein interactions can be easily extracted as shown in Figure 3 (d).

Figure 4 (a) Histogram of sentence length in the test set after simplification. (b) Histogram of length decrease of sentences in the test set by employing simplification.



4 Results

A corpus named Corpus I was constructed based on the GENIA corpus [24]. GENIA is a collection of 2000 research abstracts selected from the search results of MEDLINE database using keywords (MESH terms) *"human, blood cells and transcription factors"*. All these abstracts were then split into sentences and those containing more than two protein names and at least one interaction keyword were kept. Altogether 3533 sentences were left and 2500 sentences were sampled to build Corpus I.

We performed 10-fold cross validation on our method. The corpus I was randomly split into the training set and the test set at the ration of 9:1. The test set consists of 250 sentences and the remaining 2250 sentences were used as the training set. The experiments were conducted ten times (i.e Experiment 1, 2, ..., 10 in Table 1) with different training and test data each round.

Figure 4 (a) illustrates the distribution of the sentence length in the test set after sentence simplification. Here a protein name consisting of several words is considered as one word. Figure 4 (b) shows the decrease of sentence length in the test set by employing sentence simplification. It can be observed that sentence simplification can effectively eliminate 1 to 7 words for most sentences and the sentence length is reduced to the range of 20-40.

The average processing speed on Itanium-1 model Linux server equipped with 733Mhz processor and 4 GB RAM was 0.23s per sentence.

The results reported in this paper are based on the values of TP (true positive), FP (false positive), and FN (false negative). TP is the number of correctly extracted interactions. (TP+FN) is the number of all interactions in the test set and (TP+FP) is the number of all extracted interactions. F-score is computed using the formula below:

$$\text{F-score} = \frac{2 \cdot \text{Recall} \cdot \text{Precision}}{\text{Recall} + \text{Precision}} \tag{3}$$

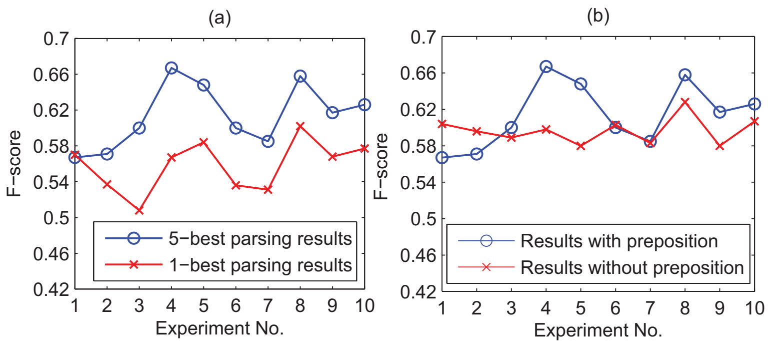
where Recall is defined as TP/(TP+FN) and Precision is defined as TP/(TP+FP).

Table 1 shows the evaluation results of 10-fold cross-validation where the average F-score value obtained is 61.5% with the balanced recall and precision values.

Table 1 Results of 10-fold cross-validation

Experiment	TP+FN	TP	NP	Recall (%)	Precision (%)	F-Score (%)
1	367	207	156	56.4	57.0	56.7
2	394	220	156	55.8	58.5	57.1
3	386	241	176	62.4	57.8	60.0
4	400	268	136	67.0	66.3	66.7
5	427	278	153	65.1	64.5	64.8
6	391	234	155	59.8	60.2	60.0
7	371	223	168	60.1	57.0	58.5
8	369	252	145	68.3	63.5	65.8
9	385	230	131	59.7	63.7	61.7
10	390	244	146	62.6	62.6	62.6
overall	3880	2397	1522	61.8	61.2	61.5

Figure 5 (a) Comparisons between results obtained from 1-best v.s. 5-best parsing results. (b) Comparisons between results with and without the preposition information.



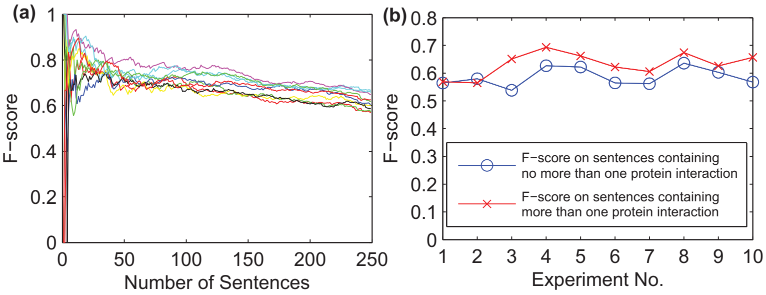
4.1 Results based on the 5-best parsing results

Instead of using the best semantic parsing result, we performed protein-protein interaction extraction based on the 5-best parsing results as mentioned in section 3. Figure 5 (a) illustrated the F-scores obtained from the 1-best or 5-best parsing results. It can be seen that using the 5-best parsing results, the relative improvement on F-score is 0 to 9%, and the average improvement on F-score is 5.58%. An example has been given in Figure 3 (c). This reveals that the best parsing result does not always present correct semantic information.

4.2 Including prepositions in the annotation

As mentioned in section 3, two types of annotations were provided for the training data set. Figure 5 (b) shows the results generated by the HVS model trained without or with the preposition information. It is observed that by including the preposition information, the average improvement on F-score is 1.71%. This gives positive support on our hypothesis that preposition information do play an impor-

Figure 6 (a) Performance on sentences with increasing length. (b) Comparisons between results on sentences containing no more than one protein-protein interaction and those on sentences containing more than one protein-protein interaction.



tant role on revealing the underlying semantic information of the sentence.

4.3 Results based on the sentence complexity

To analyze the ability of the HVS model in extracting information from syntactically complex sentences, we sorted the sentences in the test data by their length in ascending order. The rationale behind this is that in general, sentences with more words exhibit more complex syntactic structures. By adding sentences gradually, Figure 6 (a) illustrates the performances on the ten-round test. It can be observed that the performance of the HVS model only drops slightly when the test sentences become more complex. Overall, the HVS model gives quite stable performance.

We also measured the performance on the sentences containing only one protein-protein interaction and the sentences containing more than one interaction separately. It can be observed from Figure 6 (b) that F-score on sentences containing more than one protein-protein interaction is always higher than that on sentences containing a single protein-protein interaction for most experiments. It only drops slightly by 2% for the Experiment 2 test data. These results are contrary to our general belief that sentences containing more than one protein-protein interaction should exhibit more complex syntactic structures. One reason is that the system fails to extract one protein-protein interaction from a sentence containing a single protein-protein interaction will result in 0% in F-score. However, the system fails to extract one protein-protein interaction from a sentence containing two protein-protein interactions would result in 66.7% in F-score. These results can be further explained by our observation that for sentences containing more than one protein-protein interaction, their theme often focuses on protein-protein interaction only, so they are short in general, while for sentences containing a single protein-protein interaction, they might discuss something else rather than protein-protein interaction only, therefore the length of those sentences is normally longer.

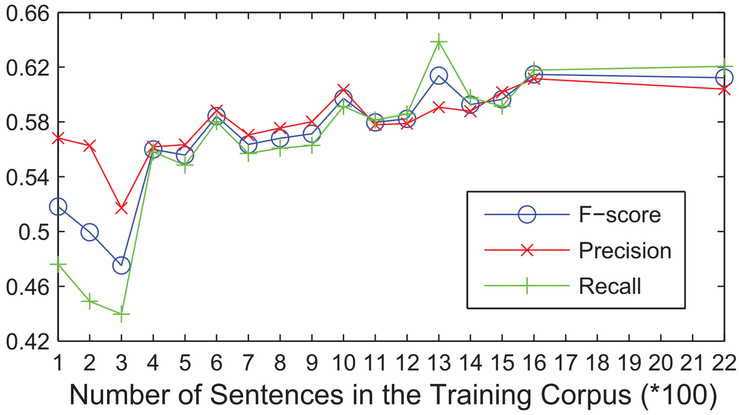
4.4 Results based on the interaction category

By analyzing the categories of protein-protein interactions in our data set (Corpus I), we found that two categories, *activate* and *attach* accounts for about 30% of

Table 2 Results based on the interaction category

Category	Recall (%)	Precision (%)	F-Score (%)
activate	66.7	68.3	67.5
attach	58.1	71.4	64.1

Figure 7 Performances of HVS model trained on the increasingly added training data.



all protein-protein interactions. Thus, the results based on these two categories are also shown here. It can be observed from Table 2 that there are slight changes in F-score when compared with the overall performance result in Table 1. It increases about 6% for the *activate* category and 2.5% for the *attach* category. One explanation for the result is that patterns about the two categories are well modeled because of enough training sentences of the two categories in the training data set.

4.5 Results based on the increasingly added training data

To explore the best performance of the HVS model, we conducted an experiment as follows. First, randomly select 100 sentences from training data (2250 sentences), use them to train an HVS model and analyze its performance based on ten-fold cross validation. Then add 100 sentences each time to build a new HVS model and analyze its performance. Figure 7 illustrates the performance on each stage. It shows that the model performance gradually improves when adding more training data. It saturates when the size of training data reaches 1600. It implies that for this particular corpus, 1600 sentences would be sufficient to train the HVS model.

4.6 Discussions

In general, it is difficult to compare performance of different approaches fairly because different corpus was employed. There are no benchmarks in biomedical text mining for protein-protein interactions. We chose the GENIA corpus as our training and test data set based on the following reasons. Firstly, protein names are fully annotated in the GENIA corpus so that we can focus on the protein-protein

Table 3 Distribution of number of sentences in each category (PPI denotes protein-protein interaction)

Result Category	Number of Sentences	Percentage (%) of Sentences
Identify all PPIs in the sentence without generating wrong PPIs	104	41.6
Identify all PPIs in the sentence, but generate wrong PPIs	44	17.6
Identify part of PPIs in the sentence without generating wrong PPIs	68	27.2
Identify part of PPIs in the sentence, and generate wrong PPIs	34	13.6
Total	250	100

interaction extraction task without being distracted by the problems in name entity identification. Secondly, the GENIA corpus has been widely used in the field of biomedical text mining especially for name entity identification. Although there is no system published so far on extracting protein-protein interactions from the GENIA corpus, we believe GENIA will be used by more and more researchers for protein-protein interactions extraction in the near future.

To investigate the reasons behind the errors in the experiments, we have analyzed the parsing results of 250 randomly selected sentences from the test data set. The parsing results are classified into four categories and frequencies in each category are presented in Table 3. It can be observed that 41.6% of sentences were correctly processed by our approach. We then proceeded to analyze the errors in the parsing results of the remaining 58.4% of sentences. Errors can be classified into three main categories as listed in Table 4. (1) Semantic parsing errors constitute the major portion of all errors. We find that the current semantic parsing method has some restrictions and causes approximately 70% of the total errors. This partially derives from the fact that some complex hierarchical structure can not be handled by our method. With these considerations, a more accurate semantic parsing method is under development. (2) Errors caused by the preprocessing procedure accounts for nearly 6% of all failures. By elaborately constructing the protein name and interaction keyword dictionary, errors in this category should be eliminated. (3) The simple extraction rules and he heuristics for result selection based on 5-best parsing paths caused about 22% errors. we currently are building a set of more comprehensive rules to solve the problem.

5 Adaptation to Changing Domains

Statistical models calculate their probability estimates based on their training data. When these models are shifted to another domain, the performance usually drops. Adaptation techniques are used to adapt a well-trained model to a novel domain. Two major approaches are commonly used: maximum *a posteriori* (MAP) estimation and discriminative training methods. For the MAP estimation methods, adaptation data are used to adjust the parameters of the model so as to maximize the likelihood of the adaptation data. Count merging and interpolation

Table 4 Classification and frequency of errors

Error Category	Reasons	Error Proportion
Semantic parsing errors Total: 106 errors (72.6%)	Negative words such as “but”, “unlike” are not considered.	5(3.4%)
	Left-branching structures can not be handled by the HVS model.	3 (2.1%)
	Interaction keywords are rarely presented in the training data.	14 (9.6%)
	Hierarchy information is generated incorrectly	84 (57.5%)
Preprocessing Errors Total: 8 errors (5.5%)	Some relevant words are removed by sentence simplification	2 (1.4%)
	Interaction keywords are not listed in the keyword dictionary.	3 (2.1%)
	Protein names are identified incorrectly.	3 (2.1%)
Postprocessing Errors Total: 32 errors (21.9%)	Wrong parsing results are chose, although true information can be found in 5-best parsing results.	12 (8.2%)
	Predefined rules for extracting protein-protein interactions from parsing results fail to extract.	20 (13.7%)

of models are the two MAP estimation methods investigated in speech recognition experiments [25]. In recent years, MAP adaptation has been successfully applied to lexicalized probabilistic context-free grammar (PCFG) models [26]. Discriminative approaches, on the other hand, aim at using the adaptation data to directly minimize the errors on the adaptation data made by the model. These techniques have been applied successfully to the task of language modeling in non-adaptation scenario [27].

Since MAP adaptation is straightforward and has been applied successfully to PCFG parsers, it has been selected for investigation in this paper. In particular, we mainly focused on one of the special forms of MAP adaptation which is interpolation between the in-domain and out-of-domain models. The following presents how to adapt the HVS model using the log-linear interpolation method [b].

5.1 Log-Linear interpolation

Log-linear interpolation has been applied to language model adaptation and has been shown to be equivalent to a constrained minimum Kullback-Leibler distance optimization problem [28].

Assume a generalized parser model $P(W,C)$ for a word sequence W and semantic concept sequence C exists with J component distributions P_j each of dimension K, then given some adaptation data W_l, the log-linear estimate of the kth compo-

[b] Experiments using linear interpolation have also been conducted but it was found that the results are worse than those obtained using log-linear interpolation.

Table 5 Performance Comparison of adaptation to Corpus II

System	Recall (%)	Precision (%)	F-score (%)
Baseline	39.9	60.1	48.0
In domain	53.4	62.6	57.6
Log-Linear	44.4	71.2	54.7

nent of P_j, $\hat{P}_j(k)$, is

$$\hat{P}_j(k) = \frac{1}{Z_\lambda} P_j(k)^{\lambda_1} \tilde{P}_j(k)^{\lambda_2} \tag{4}$$

where $P_j(k)$ is the probability of the original unadapted model, and $\tilde{P}_j(k)$ is the empirical distribution of the adaptation data defined as

$$\tilde{P}_j(k) = \frac{\sigma_j(k)}{\sum_{i=1}^{K} \sigma_j(i)} \tag{5}$$

in which $\sigma_j(k)$ is defined as the total count of the events associated with the kth component of P_j summed across the decoding of all adaptation utterances W_l. The parameters λ_1 and λ_2 were determined by optimizing the log-likelihood on the held-out data using the simplex method. The computation of Z_λ is very expensive and can usually be dropped without significant loss in performance [29].

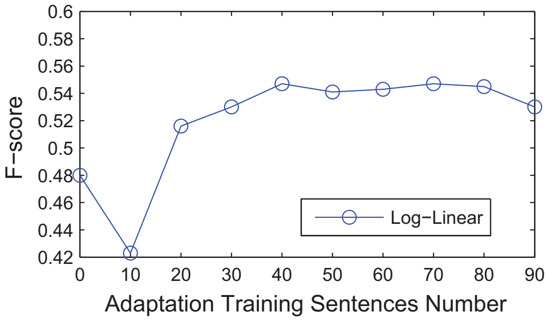
5.2 Experimental results

To justify the robustness of the HVS parser, another corpus named as Corpus II was used. Corpus II were obtained from [10]. The initial corpus consists of 1203 sentences which are accompanied with their respective protein interaction information. All sentences were examined manually to ensure the correctness of protein-protein interactions. After cleaning up the sentences which do not contain protein interaction information, 800 sentences were kept. Note that Corpus II is constructed from the first 50 biomedical papers downloaded from the Internet with the keyword "*protein-protein interaction*". Corpus I and Corpus II are disjoint sets. Corpus I, a collection of abstracts, and Corpus II, a set of sentences from full papers might comprise different writing styles.

The baseline HVS model was trained on data from Corpus I and was later adapted using a small amount of adaptation data from Corpus II. Table 5 lists the recall, precision, and F-score obtained when tested on data from Corpus II (100 sentences). The "Baseline" results were obtained using the HVS model trained on data from Corpus I without adaptation. The "In domain" results were obtained using the HVS model trained solely on the Corpus II sentences. The "Log-Linear" row shows the performance using the log-linear interpolation based adaptation of the baseline model using 90 randomly selected adaptation sentences from Corpus II.

Figure 8 shows the parser performance versus the number of adaptation sentences used. It can be observed that the F-score value increases when increasingly adding more adaptation data from Corpus II. The parser performance almost saturates when the number of adaptation utterances reaches 40. The performance

Figure 8 F-score vs amount of adaptation training data.



however degrades when the number of adaptation utterances exceeds 80, possibly due to model overtraining. For this particular application, we conclude that just 80 adaptation utterances would be sufficient to adapt the baseline model to give comparable results to the in-domain model. Overall, we found that directly moving a HVS model trained on data from Corpus I to Corpus II resulted in a 10% absolute drop in F-score. However, when adaptation was applied using only 40 adaptation sentences, the loss of concept accuracy was dramatically restored. Specifically, using log-linear adaptation, the out-of-domain F-score of 48.0% was restored to 54.7%, which is not far from the in-domain F-score of 57.6%.

6 Conclusion

In this paper, we have presented an approach based on the HVS model to automatically extract protein-protein interactions from unstructured text sources. The approach can generate satisfactory performance measured in recall and precision. We have also investigated the ability of the HVS model to be adapted to another domain. The experimental results give positive support that the purely data-driven extraction system is robust and can be readily adapted to a new domain. Our work may provide a useful supplement to manually created resources in established public databases. In future work we will work on the enhancement of our approach in order to improve the extraction accuracy.

References and Notes

1 GD. Bader, D. Betel, and CW. Hogue. BIND: the Biomolecular Interaction Network Database. *Nucleic Acids Research*, 31(1):248–250, 2003.

2 H. Hermjakob, L. Montecchi-Palazzi, and C. Lewington. IntAct: an open source molecular interaction database. *Nucleic Acids Research.*, 1(32(Database issue)):452–5, 2004.

3 C. von Mering, LJ. Jensen, B. Snel, SD. Hooper, and M. Krupp. STRING: known and predicted protein-protein associations, integrated and transferred across organisms. *Nucleic Acids Research*, 33(Database issue):433–7, 2005.

4 Miguel A. Andrade and Alfonso Valencia. Automatic extraction of keywords from scientific text: application to the knowledge domain of protein families . *Bioinformatic*, 14(7):600–607, 1998.

5 Edward M. Marcotte, Ioannis Xenarios, and David Eisenberg. Mining literature for protein-protein interactions. *Bioinformatics*, 17(4):359–363, 2001.

6 B. Stapley and G. Benoit. Biobibliometrics: information retrieval and visualization from co-occurrences of gene names in Medline abstracts. In *Proceedings of the Pacific Symposium on Biocomputing*, pages 529–540, Hawaii, U.S.A, 2000.

7 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.

8 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.

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

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

11 Craven Mark and Kumlien Johan. Constructing biological knowledge bases by extracting information from text sources. In *Proceedings of the 7th International Conference on Intelligent Systems for Molecular Biology*, pages 77–86, Heidelberg, Germany, 1999.

12 J. Pustejovsky, J. Castano, J. Zhang, M. Kotecki, and B. Cochran. Robust Relational Parsing Over Biomedical Literature: Extracting Inhibit Relations. In *Proceedings of the Pacific Symposium on Biocomputing.*, pages 362–373, Hawaii, U.S.A, 2002.

13 A. Yakushiji, Y. Tateisi, Y. Miyao, and J. Tsujii. Event extraction from biomedical papers using a full parser. In *Proceedings of the Pacific Symposium on Biocomputing*, volume 6, pages 408–419, Hawaii, U.S.A, 2001.

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

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

16 Deyu Zhou, Yulan He, and Chee Keong Kwoh. Extracting Protein-Protein Interactions from the Literature using the Hidden Vector State Model. In *International Workshop on Bioinformatics Research and Applications*, pages 718–725, Reading, UK, 2006.

17 Lynette Hirschman, Alexander Yeh, Christian Blaschke, and Alfonso Valencia. Overview of BioCreAtIvE: critical assessment of information extraction for biology. *BMC Bioinformatics*, 6(Suppl 1), 2004.

18 G. Bakiri and T. G. Dietterich. Achieving high-accuracy text-to-speech with machine learning. *Data Mining Techniques in Speech Synthesis*, 2002.

19 L. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

20 Andrew McCallum, Dayne Freitag, and Fernando Pereira. Maximum entropy Markov models for information extraction and segmentation. In *Proc. 17th International Conf. on Machine Learning*, pages 591–598. Morgan Kaufmann, San Francisco, CA, 2000.

21 John Lafferty, Andrew McCallum, and Fernando Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *In International Conference Machine Learning*, San Francisco, CA, 2001. Morgan Kaufmann, 2001.

22 Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

23 Thomas G. Dietterich. Machine learning for sequential data: A review. In *Proceedings of the Joint IAPR International Workshop on Structural, Syntactic, and Statistical Pattern Recognition*, pages 15–30, London, UK, 2002. Springer-Verlag.

24 JD. Kim, T. Ohta, Y. Tateisi, and J Tsujii. GENIA corpus–semantically annotated corpus for bio-textmining. *Bioinformatics*, 19(Suppl 1):i180–2, 2003.

25 R. Iyer, M. Ostendorf, and H. Gish. Using out-of-domain data to improve in-domain language models. *IEEE Signal Processing Letters*, 4(9):221–223, 1997.

26 B. Roark and M. Bacchiani. Supervised and unsupervised PCFG adaptation to novel domains. In *Proceedings of the joint meeting of the North American Chapter of the Association for Computational Linguistics and the Human Language Technology Conference*, pages 126–133, Edmonton, Canada, May 2003.

27 B. Roark, M. Saraclar, and M. Collins. Corrective language modeling for large vocabulary asr with the perceptron algorithm. *Proceedings of ICASSP*, pages 749–752, 2004.

28 D. Klakow. Log-linear interpolation of language models. In *Proc. of Intl. Conf. on Spoken Language Processing*, Sydney, Australia, Nov. 1998.

29 S. Martin, A. Kellner, and T. Portele. Interpolation of stochastic grammar and word bigram models in natural language understanding. In *Proc. of Intl. Conf. on Spoken Language Processing*, Beijing, China, Oct. 2000.