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

Deyu Zhou* and Yulan He

Informatics Research Centre,
University of Reading,
3rd Floor, Philip Lyle Building,
Whiteknights, Reading RG6 6BX, UK
E-mail: d.zhou@reading.ac.uk
E-mail: y.he@reading.ac.uk
*Corresponding author

Chee Keong Kwoh

School of Computer Engineering,
Nanyang Technological University,
Nanyang Avenue, 639798 Singapore
E-mail: asckkwoh@ntu.edu.sg

Abstract: A major challenge in text mining for biomedicine is automatically extracting protein-protein interactions from the vast amount of biomedical literature. We have constructed an information extraction system based on the Hidden Vector State (HVS) model for protein-protein interactions. The HVS model can be trained using only lightly annotated data whilst simultaneously retaining sufficient ability to capture the hierarchical structure. When applied in extracting protein-protein interactions, 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.

Keywords: information extraction; Hidden Vector State model; Protein-Protein Interactions; PPIs; bioinformatics.

Reference to this paper should be made as follows: Zhou, D., He, Y. and Kwoh, C.K. (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 PhD candidate in the Informatics Research Centre at the University of Reading. His interests are statistical methods for mining knowledge from texts and biomedical data mining. He received BS Degree in Mathematics in 2000 and ME Degree in Computer Science from Nanjing University in 2003.

Yulan He is a Lecturer in the Informatics Research Centre, the School of Business, the University of Reading, UK. She obtained her BASc (1st class

Honours) and MEng in 1997 and 2001 respectively, both from Nanyang Technological University, Singapore. Following graduation in 1997 she worked as a System Analyst in the Development Bank of Singapore. In March 2000 she took up a position as a Senior Engineer in Kent Ridge Digital Labs (now Institute for Infocomm Research), Singapore. From October 2001 to September 2004, she was a PhD candidate in Cambridge University Engineering Department, UK and obtained her PhD Degree in 2004. Between 2004 and 2007, she was an Assistant Professor with the School of Computer Engineering at the Nanyang Technological University, Singapore. Her current research interests include 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 PhD 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 (PPIs) 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 an understanding of living cells as complex machines and disease processes, and provides targets for effective drug designs. Although many databases, such as BIND (Bader et al., 2003), IntAct (Hermjakob et al., 2004) and STRING (von Mering et al., 2005), have been built to store PPI information, constructing such databases is time-consuming and needs an immense amount of manual effort to ensure the correctness of data. To date, a vast quantity of knowledge about PPIs still hides in biomedical literature. As a result, automatically extracting this information from biomedical text holds the promise of easily discovering large amounts of biological knowledge in computer-accessible form.

In the earlier stages of this field of study, statistical methods (Andrade and Valencia, 1998; Marcotte et al., 2001) were employed to search abstracts or sentences which may describe PPIs based on the co-occurrence of protein names. Following that, other approaches (Stapley and Benoit, 2000; Donaldson et al., 2003) focused on detecting proteins pairs and determining the relations between them based on some probability scores. Obviously, these approaches cannot achieve a satisfactory performance because they ignore sentence structures which play an important role in expressing PPIs.

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 (Ono et al., 2001; Blaschke and Valencia, 2002; Huang et al., 2004) rely on a set of predefined or automatically generated patterns to extract PPIs. For example, Ono et al. (2001) 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*. Blaschke and Valencia (2002) 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 (Mark and Johan, 1999; Pustejovsky et al., 2002) 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 (Yakushiji et al., 2001; Temkin and Gilder, 2003) deal with the structure of an entire sentence and therefore are potentially more accurate. Yakushiji et al. (2001) 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% (Temkin and Gilder, 2003).

In this paper, we propose an approach based on the HVS model to automatically extract PPIs from biomedical literature. The HVS model has been successfully applied to discover semantic information in spoken utterances (He and Young, 2005). 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 PPIs extraction (Zhou et al., 2006).

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% (Temkin and Gilder, 2003), 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 PPIs.[1]

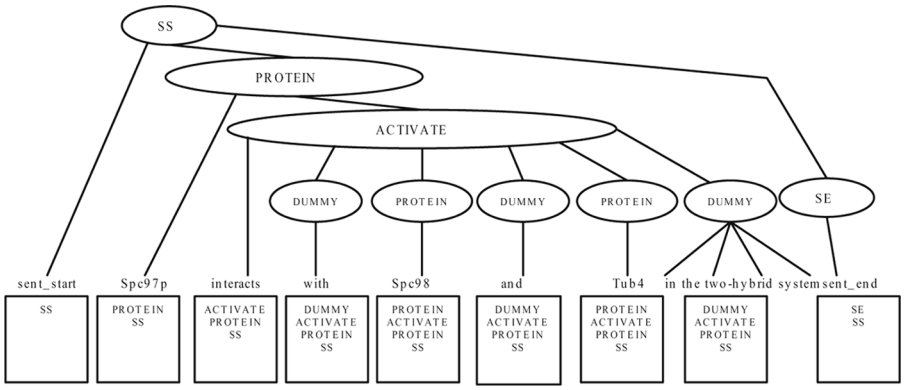
The rest of the paper is organised as follows: In the next section, we will briefly describe the HVS model and how it can be used to extract PPIs from biomedical literature. In Section 3, we present the overall structure of the extraction system and its components. Experimental results are discussed in Section 4. Adaptation methods

and results are presented in Section 5. Finally, Section 6 concludes the paper and gives future directions.

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 $S = \langle s_1, s_2, \cdots, s_n \rangle$.

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



Existing statistical approaches to this problem include sliding-window methods (Bakiri and Dietterich, 2002), hidden Markov models (Rabiner, 1989), maximum entropy Markov models (McCallum et al., 2000), conditional random fields (Lafferty et al., 2001), graph transformer networks (LeCun et al., 1998) etc. [for a review, see Dietterich, 2002].

While the aforementioned approaches require fully annotated training corpora for model parameter estimation in general, we propose a HVS model (He and Young, 2005) 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)

$$\begin{array}{l}\text{PROTEIN=Spc97}\\ \text{PROTEIN.ACTIVATE=interacts}\\ \text{PROTEIN.ACTIVATE.PROTEIN=Spc98}\\ \text{PROTEIN.ACTIVATE.PROTEIN=Tub4.}\end{array} \tag{1}$$

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

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

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., $\mathbf{c}_t = [c_t[1], c_t[2], \ldots, c_t[D_t]]$, $c_t[1] = c_{w_t}$ 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:

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

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 He and Young (2005).

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 identified based on a pre-constructed biological term dictionary. After that, each sentence is parsed by the HVS semantic parser. Finally, PPIs 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.

Figure 2 System architecture

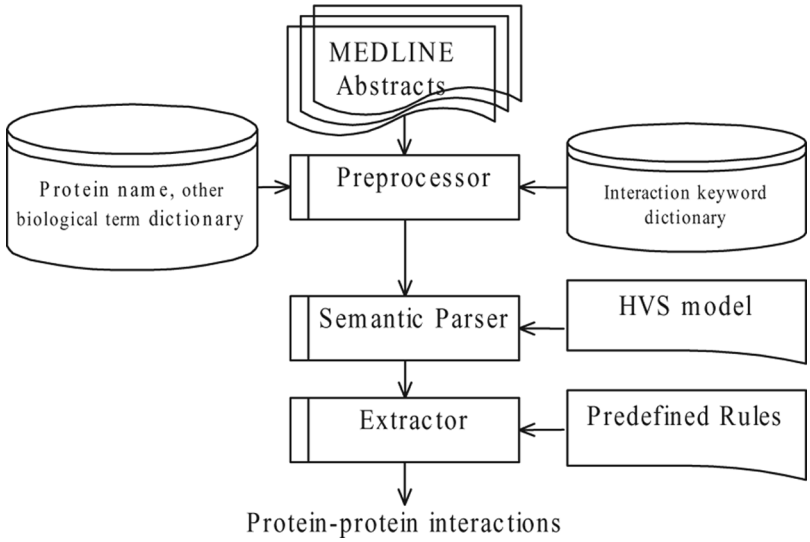
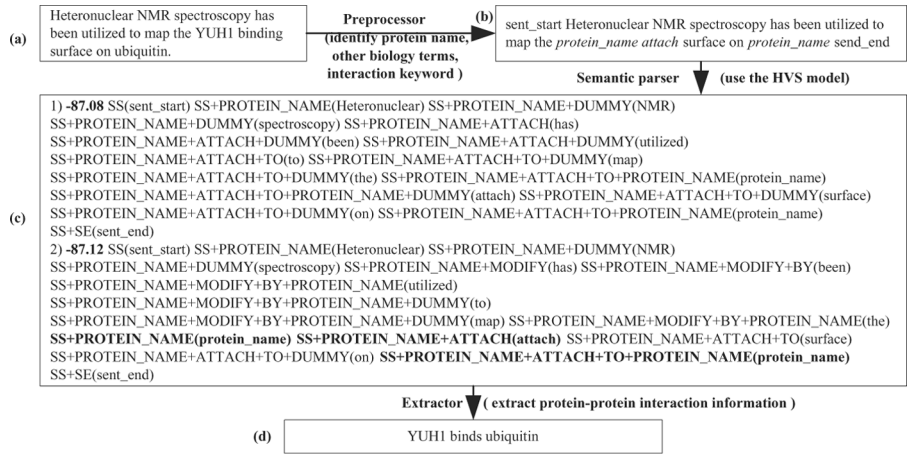


Figure 3 An example of a procedure for information extraction using the HVS model



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

To extract PPIs 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 Temkin and Gilder (2003). 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 PPIs. 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 PPI, 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-8andSKR-10inyeasttwo − hybridsystem.
$Annotation$: 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 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, PPIs 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 PPIs from this parsing output. Otherwise, the best parsing result will still be used. Figure 3(c) and (d) illustrates the application of this rule where the PPI 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+PROTEIN_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, PPIs can be easily extracted as shown in Figure 3(d).

4 Results

A corpus named Corpus I was constructed based on the GENIA corpus (Kim et al., 2003). 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.

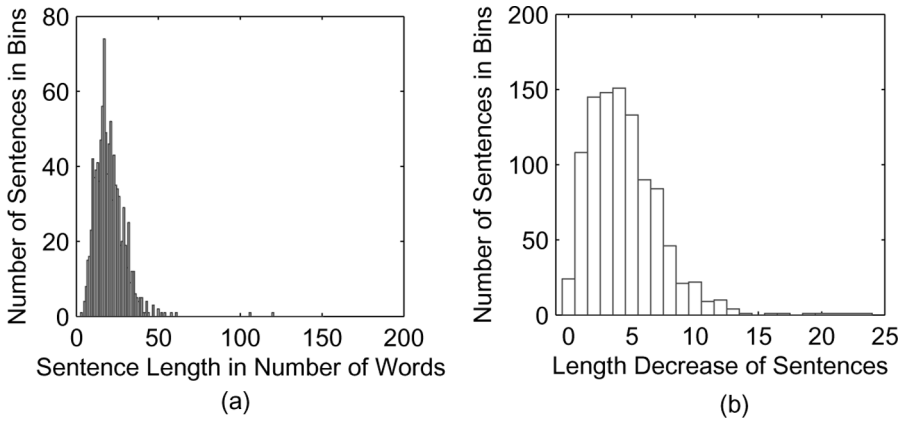
We performed 10-fold cross validation on our method. The corpus I was randomly split into the training set and the test set at the ration of $9:1$. The test set consists of 250 sentences and the remaining 2250 sentences were used as the training set. The experiments were conducted ten times (i.e., Experiment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

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 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 1–7 words for most sentences and the sentence length is reduced to the range of 20–40.

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



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

The results reported in this paper are based on the values of True Positive (TP), False Positive (FP), and False Negative (FN). 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.

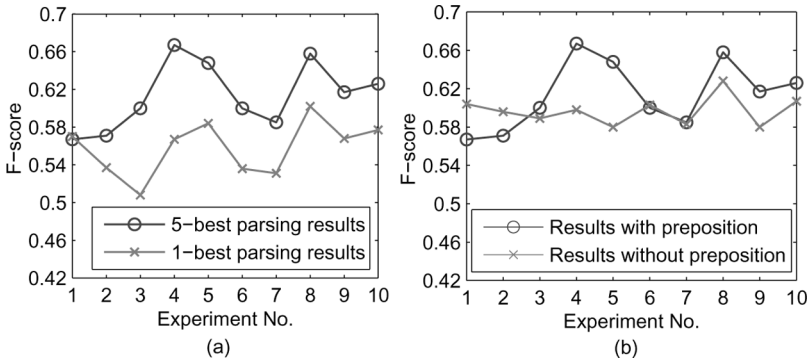
4.1 Results based on the 5-best parsing results

Instead of using the best semantic parsing result, we performed PPI 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 important role on revealing the underlying semantic information of the sentence.

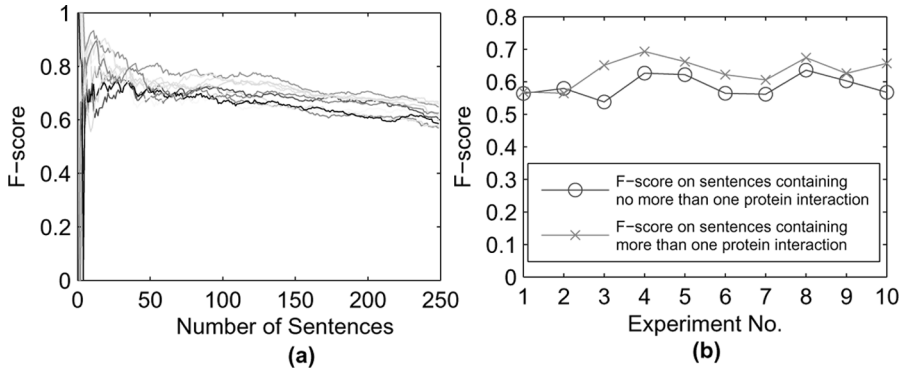
Figure 5 (a) Comparisons between results obtained from 1-best v.s. 5-best parsing results and (b) comparisons between results with and without the preposition information



4.3 Results based on the sentence complexity

To analyse 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.

Figure 6 (a) Performance on sentences with increasing length and (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



We also measured the performance on the sentences containing only one PPI 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 PPI is always higher than that on sentences containing a single PPI 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 PPI should exhibit more complex syntactic structures. One reason is that the system fails to extract one PPI from

a sentence containing a single PPI will result in 0% in F-score. However, the system fails to extract one PPI from a sentence containing two PPIs would result in 66.7% in F-score. These results can be further explained by our observation that for sentences containing more than one PPI, their theme often focuses on PPI only, so they are short in general, while for sentences containing a single PPI, they might discuss something else rather than PPI only, therefore the length of those sentences is normally longer.

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

4.4 Results based on the interaction category

By analysing the categories of PPIs in our data set (Corpus I), we found that two categories, *activate* and *attach* accounts for about 30% of all PPIs. 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 modelled because of enough training sentences of the two categories in the training data set.

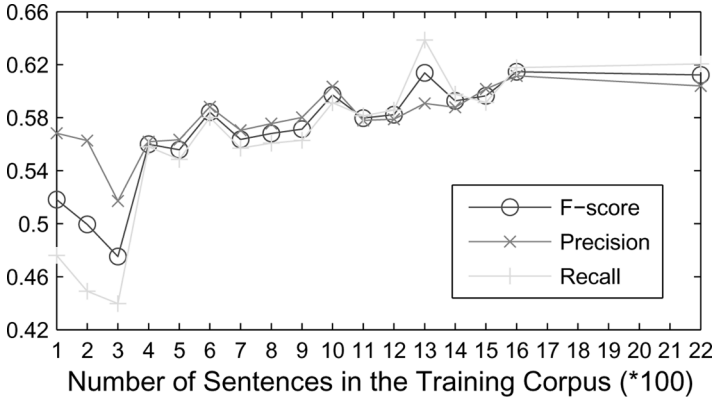
4.5 Results based on the increasingly added training data

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 analyse its performance based on ten-fold cross validation. Then add 100 sentences each time to build a new HVS model and analyse 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 PPIs. 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 PPI 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 PPIs from the GENIA corpus, we believe GENIA will be used by more and more researchers for PPIs extraction in the near future.

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



To investigate the reasons behind the errors in the experiments, we have analysed 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 analyse 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.

Table 3 Distribution of number of sentences in each category (PPI denotes Protein-Protein Interaction)

Result category	*No. 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*

- 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.
- 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.

- 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.

Table 4 Classification and frequency of errors

Error category	*Reasons*	*Error proportion*
Semantic parsing errors	Negative words such as 'but', 'unlike' are not considered	5 (3.4%)
Total: 106 errors (72.6%)	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	Some relevant words are removed by sentence simplification	2 (1.4%)
Total: 8 errors (5.5%)	Interaction keywords are not listed in the keyword dictionary	3 (2.1%)
	Protein names are identified incorrectly.	3 (2.1%)
Postprocessing errors	Wrong parsing results are chose, although true information can be found in 5-best parsing results	12 (8.2%)
Total: 32 errors (21.9%)	Predefined rules for extracting PPIs from parsing results fail to extract	20 (13.7%)

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 maximise the likelihood of the adaptation data. Count merging and interpolation of models are the two MAP estimation methods investigated in speech recognition experiments (Iyer et al., 1997). In recent years, MAP adaptation has been successfully applied to lexicalised Probabilistic Context-Free Grammar (PCFG) models (Roark and Bacchiani, 2003). Discriminative approaches, on the other hand, aim at using the adaptation data to directly minimise the errors on the adaptation data made by the model. These techniques have been applied successfully to the task of language modelling in non-adaptation scenario (Roark et al., 2004).

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.[2]

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 optimisation problem (Klakow, 1998).

Assume a generalised 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 component 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 optimising 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 (Martin et al., 2000).

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 Huang et al. (2004). The initial corpus consists of 1203 sentences which are accompanied with their respective PPI. All sentences were examined manually to ensure the correctness of PPIs. 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*". Corpuses I and 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.

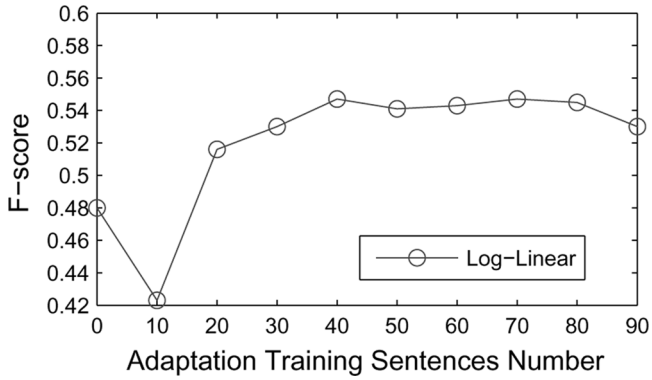
Figure 8 shows the parser performance vs. the number of adaptation sentences used. It can be observed that the F-score value increases when increasingly adding more

adaptation data from Corpus II. The parser performance almost saturates when the number of adaptation utterances reaches 40. The performance 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 Corpuses I to 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%.

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

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



6 Conclusion

In this paper, we have presented an approach based on the HVS model to automatically extract PPIs 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

Andrade, M.A. and Valencia, A. (1998) 'Automatic extraction of keywords from scientific text: application to the knowledge domain of protein families', *Bioinformatic*, Vol. 14, No. 7, pp.600–607.

Bader, G.D., Betel, D. and Hogue, C.W. (2003) 'BIND: the biomolecular interaction network database', *Nucleic Acids Research*, Vol. 31, No. 1, pp.248–250.

Bakiri, G. and Dietterich, T.G. (2002) 'Achieving high-accuracy text-to-speech with machine learning', in Damper, B. (Ed.): *Data Mining in Speech Synthesis*, Chapman and Hall.

Blaschke, C. and Valencia, A. (2002) 'The frame-based module of the SUISEKI information extraction system', *IEEE Intelligent Systems*, Vol. 17, No. 2, pp.14–20.

Dietterich, T.G. (2002) 'Machine learning for sequential data: a review', *Proceedings of the Joint IAPR International Workshop on Structural, Syntactic, and Statistical Pattern Recognition*, Springer-Verlag, London, UK, pp.15–30.

Donaldson, I., Martin, J., de Bruijn, B. and Wolting, C. (2003) 'PreBIND and textomy-mining the biomedical literature for protein-protein interactions using a support vector machine', *BMC Bioinformatics*, Vol. 4, No. 11.

He, Y. and Young, S. (2005) 'Semantic processing using the hidden vector state model', *Computer Speech and Language*, Vol. 19, No. 1, pp.85–106.

Hermjakob, H., Montecchi-Palazzi, L. and Lewington, C. (2004) 'IntAct: an open source molecular interaction database', *Nucleic Acids Research*, Vol. 1, No. 32, Database Issue, pp.452–455.

Hirschman, L., Yeh, A., Blaschke, C. and Valencia, A. (2004) 'Overview of BioCreAtIvE: critical assessment of information extraction for biology', *BMC Bioinformatics*, Vol. 6, Suppl. 1, p.S1.

Huang, M., Zhu, X. and Hao, Y. (2004) 'Discovering patterns to extract protein-protein interactions from full text', *Bioinformatics*, Vol. 20, No. 18, pp.3604–3612.

Iyer, R., Ostendorf, M. and Gish, H. (1997) 'Using out-of-domain data to improve in-domain language models', *IEEE Signal Processing Letters*, Vol. 4, No. 9, pp.221–223.

Kim, J.D., Ohta, T., Tateisi, Y. and Tsujii, J. (2003) 'GENIA corpus-semantically annotated corpus for bio-textmining', *Bioinformatics*, Vol. 19, Suppl. 1, pp.i180–182.

Klakow, D. (1998) 'Log-linear interpolation of language models', *Proc. Intl. Conf. on Spoken Language Processing*, Sydney, Australia, November, pp.1695–1698.

Lafferty, J., McCallum, A. and Pereira, F. (2001) 'Conditional random fields: probabilistic models for segmenting and labeling sequence data', *International Conference Machine Learning*, Morgan Kaufmann, San Francisco, CA. pp.282–289.

LeCun, Y., Bottou, L., Bengio, Y. and Haffner, P. (1998) 'Gradient-based learning applied to document recognition', *Proceedings of the IEEE*, Vol. 86, No. 11, pp.2278–2324.

Marcotte, E.M., Xenarios, I. and Eisenberg, D. (2001) 'Mining literature for protein-protein interactions', *Bioinformatics*, Vol. 17, No. 4, pp.359–363.

Mark, C. and Johan, K. (1999) 'Constructing biological knowledge bases by extracting information from text sources', *Proceedings of the 7th International Conference on Intelligent Systems for Molecular Biology*, Heidelberg, Germany, pp.77–86.

Martin, S., Kellner, A. and Portele, T. (2000) 'Interpolation of stochastic grammar and word bigram models in natural language understanding', *Proc. Intl. Conf. on Spoken Language Processing*, Beijing, China, October, Vol. 1, pp.234–237.

McCallum, A., Freitag, D. and Pereira, F. (2000) 'Maximum entropy Markov models for information extraction and segmentation', *Proc. 17th International Conf. on Machine Learning*, Morgan Kaufmann, San Francisco, CA, pp.591–598.

Ono, T., Hishigaki, H., Tanigam, A. and Takagi, T. (2001) 'Automated extraction of information on protein-protein interactions from the biological literature', *Bioinformatics*, Vol. 17, No. 2, pp.155–161.

Pustejovsky, J., Castano, J., Zhang, J., Kotecki, M. and Cochran, B. (2002) 'Robust relational parsing over biomedical literature: extracting inhibit relations', *Proceedings of the Pacific Symposium on Biocomputing*, Hawaii, USA, pp.362–373.

Rabiner, L.R. (1989) 'A tutorial on hidden Markov models and selected applications in speech recognition', *Proceedings of the IEEE*, Vol. 77, No. 2, pp.257–286.

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

Roark, B., Saraclar, M. and Collins, M. (2004) 'Corrective language modeling for large vocabulary asr with the perceptron algorithm', *Proceedings of ICASSP*, Montreal, Quebec, Canada, pp.749–752.

Stapley, B. and Benoit, G. (2000) 'Biobibliometrics: information retrieval and visualization from co-occurrences of gene names in Medline abstracts', *Proceedings of the Pacific Symposium on Biocomputing*, Hawaii, USA, pp.529–540.

Temkin, J.M. and Gilder, M.R. (2003) 'Extraction of protein interaction information from unstructured text using a context-free grammar', *Bioinformatics*, Vol. 19, No. 16, pp.2046–2053.

von Mering, C., Jensen, L.J., Snel, B., Hooper, S.D. and Krupp, M. (2005) 'STRING: known and predicted protein-protein associations, integrated and transferred across organisms', *Nucleic Acids Research*, Vol. 33, Database Issue, pp.433–437.

Yakushiji, A., Tateisi, Y., Miyao, Y. and Tsujii, J. (2001) 'Event extraction from biomedical papers using a full parser', *Proceedings of the Pacific Symposium on Biocomputing*, Hawaii, USA, Vol. 6, pp.408–419.

Zhou, D., He, Y. and Kwoh, C.K. (2006) 'Extracting protein-protein interactions from the literature using the hidden vector state model', *International Workshop on Bioinformatics Research and Applications*, Reading, UK, pp.718–725.ν

Notes

[1]BioCreAtIvE challenge (Hirschman et al., 2004) 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 PPIs from text is targeted as a main task in BioCreAtIvE II 2006 and evaluation corpora are still not released to the public.

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